

BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC NHA TRANG

KHOA CÔNG NGHỆ THÔNG TIN

~~~~~~\*~~~~~~



**BÁO CÁO**

Lập trình nhúng

|  |  |  |
| --- | --- | --- |
| *Giáo viên hướng dẫn* | : | *Mai Cường Thọ* |
| *Lớp* | : | *61.CNTT-1* |
| *Tên sinh viên* | : | *Phan Trần Hữu Phúc* |
|  |  |  |

NHA TRANG – 2021

**Mục lục**

**Chủ đề 1: Lập trình với Arduino**

[Bài 1. Nháy Led 3](#_Toc90620771)

[Bài 2. Nhấn nút bật LED 4](#_Toc90620772)

[Bài 3. Cảm biến nhiệt độ TMP36 5](#_Toc90620773)

[Bài 4. Led sáng dần 6](#_Toc90620774)

[Bài 5. Led 7 màu 8](#_Toc90620775)

[Bài 6. Điều chỉnh độ sáng Led bằng chiết áp 9](#_Toc90620776)

[Bài 7. Led 7 đoạn chung cực âm 11](#_Toc90620777)

[Bài 8. Led 7 đoạn chung cực âm với bộ chuyển đổi BCD 14](#_Toc90620778)

[Bài 9. Sáng 8 Led tuần tự với IC 74HC595 16](#_Toc90620779)

[Bài 10. Led Matrix 18](#_Toc90620780)

[Bài 11. Cảm biến khoảng cách 21](#_Toc90620781)

[Bài 12. Điều khiển Motor DC 23](#_Toc90620782)

[Bài 13. Hiển thị LCD (không sử dụng thư viện) 26](#_Toc90620783)

[Bài 14. Bật tắt bóng đèn với cảm biến độ sáng 30](#_Toc90620784)

[Bài 15. Điều khiển hiển thị từ 0.0 – 9.9 bằng 2 Led 7 đoạn 31](#_Toc90620785)

[Bài 16. Điều khiển 3 LED bằng Remote hồng ngoại 35](#_Toc90620786)

[Bài 17. Keypad 4x4, hiển thị giá trị trên LCD (sử dụng thư viện Keypad.h) 38](#_Toc90620787)

[Bài 18. Điều chỉnh đèn và quạt theo độ sáng và nhiệt độ môi trường. 41](#_Toc90620788)

[Bài 19. Điều khiển Led RGB bằng Remote hồng ngoại 44](#_Toc90620789)

[Bài 20. Cảm biến nhiệt độ - độ ẩm DHT11 (sử dụng thư viện DHT.h) 47](#_Toc90620790)

**Chủ đề 2: Lập trình với STM32**

[Bài 1. Nháy Led -STM32 51](#_Toc90620791)

[Bài 2. Nháy Led Trái Tim - STM32 52](#_Toc90620792)

[Bài 3. Nút Bấm - STM32 59](#_Toc90620793)

[Bài 4. Lập Trình Ngắt - STM32 60](#_Toc90620794)

[Bài 5. Màn hình LCD - STM32 62](#_Toc90620795)

[Bài 6. Điều kiển Motor - STM32 (lập trình timer cơ bản) 66](#_Toc90620796)

[Bài 7. Chuyển đổi tín hiệu analog nhận được từ chiết áp sang tín hiệu số (ADC) 70](#_Toc90620797)

Chủ đề 1: Lập trình với Arduino

# Bài 1. Nháy Led

## Mô tả

Bài này, lập trình để thực hiện việc tự động nháy đèn led sau khoảng thời gian 500 mili giây. Led được đấu vào cổng 13 của bo mạch Arduino.

## Sơ đồ thiết kế

Hình 1.Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Led: D1 (màu xanh nước biển)
* Điện trở: R1 (100 Ω)
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| void setup() {  pinMode(13, OUTPUT);  }  void loop() {  digitalWrite(13, HIGH); // Bật led  delay(500); // Đợi 500 mili giây thì tắt led  digitalWrite(13, LOW); // Tắt led  delay(500); // Đợi 500 mili giây thì bật led  } |

# Bài 2. Nhấn nút bật LED

## Mô tả

Bài này, thực hiện lập trình nhấn nút thì bật đèn LED, thả nút ra thì LED tắt. LED được đấu ở chân thứ 13, nút bấm được đấu ở chân thứ 2.

## Sơ đồ thiết kế

Hình 2.Sơ đồ mạch Arduino

## Đặc điểm linh kiện

* Led: D1 (màu vàng)
* Điện trở: R1 (100 Ω), R2 (10k Ω)
* Nút bấm
* Mạch Arduino UNO

## Mã lệnh chính

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

# Bài 3. Cảm biến nhiệt độ TMP36

## Mô tả

Bài này, thực hiện lập trình Arduino với cảm biến nhiệt độ TMP 36 đấu ở chân A0, nếu nhiệt độ lớn hơn 37 độ thì đèn Led sáng, ngược lại thì không. Led được đấu ở chân ~10.

## Sơ đồ thiết kế

Hình 3. Sơ đồ mạch Arduino

Hình 4.Sơ đồ mạch Arduino

## Đặc điểm linh kiện

* Led: D1 (màu đỏ)
* Điện trở: R1 (100Ω)
* Cảm biến nhiệt TMP 36: có thể đo được nhiệt độ từ **-40 =>125** độ C. Khi Arduino làm việc nó sẽ trả tín hiệu tương tự này về giá trị từ 0-1023.
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| int temperature = 0;  int temp[5]; // Lấy ra những giá trị nhiệt độ tốt.  void **setup**() {  pinMode(10, OUTPUT);  }  void **loop**() {  // Chuyển đổi giá trị đọc được từ chân A0 thành nhiệt độ tương ứng  temp[0] = map(analogRead(A0),20,358,-40,125);  delay(40); // Đợi 40 mili giây để lấy giá trị tiếp theo    temp[1] = map(analogRead(A0);,20,358,-40,125);  delay(40); // Đợi 40 mili giây để lấy giá trị tiếp theo    temp[2] = map(analogRead(A0),20,358,-40,125);  delay(40); // Đợi 40 mili giây để lấy giá trị tiếp theo    temp[3] = map(analogRead(A0),20,358,-40,125);  delay(40); // Đợi 40 mili giây để lấy giá trị tiếp theo    temp[4] = map(analogRead(A0),20,358,-40,125);    // Lấy nhiệt độ trung bình  temperature = (temp[0]+temp[1]+temp[2]+temp[3]+temp[4])/5;    if(temperature > 37) {digitalWrite(10,HIGH);} // Bật led  else {digitalWrite(10,LOW);} // Tắt led    delay(100); // Đợi 100 mili giây  } |

# Bài 4. Led sáng dần

## Mô tả

Bài này, thực hiện lập trình để đèn led sáng dần trong 1 giây, sau đó tối dần lại liên tục. Led được đấu ở chân ~9.

## Sơ đồ thiết kế

Hình 4. Sơ đồ mạch Arduino

## Đặc điểm linh kiện

* Led: D1 (màu đỏ)
* Điện trở: R1
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| int brightness = 0; // Lưu giá trị độ sáng  void **setup**() {  pinMode(9, OUTPUT);  }  void **loop**() {  for (brightness = 0; brightness <= 255; brightness += 5) {  analogWrite(9, brightness); // Ghi giá trị độ sáng vào chân 9  delay(30); // Đợi trong 30 mili giây  }  for (brightness = 255; brightness >= 0; brightness -= 5) {  analogWrite(9, brightness); // Ghi giá trị độ sáng vào chân 9  delay(30); // Đợi trong 30 mili giây  }  } |

# Bài 5. Led 7 màu

## Mô tả

Đây là bài thực hành lập trình arduino với đèn led 7 màu, led 7 màu sẽ sáng trong vòng 1 giây rồi tắt liên tục. Red color được đấu ở chân thứ 5, Green color được đấu ở chân 6, Blue color được đấu ỏ chân 3 của mạch Arduino.

## Sơ đồ thiết kế

Hình 5. Sơ đồ mạch Arduino

## Đặc điểm linh kiện

* Led RGB
* Điện trở: R1(100Ω), R2(100Ω), R3(100Ω)
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| int LEDblue=3; //Blue ở chân 3  int LEDred=5; //Red ở chân 5  int LEDgreen=6; //Green ở chân 6  int b=1000; //b ngừng trong 1000ms (1 giây)  int brightnessblue=150; //giá trị từ 0 đến 255 - định nghĩa độ sáng của màu đơn  int brightnessred=150; //giá trị từ 0 đến 255 - định nghĩa độ sáng của màu đơn  int brightnessgreen=150; //giá trị từ 0 đến 255 - định nghĩa độ sáng của màu đơn  int dark=0; //Giá trị 0 là điện áp ở 0V - do đó đèn LED tắt  // Hàm thiết lập màu  void **setColor**(int r,int g,int b){  digitalWrite(LEDred, r);  digitalWrite(LEDgreen, g);  digitalWrite(LEDblue, b);  delay(b); //Break  }  void **setup**(){  pinMode(LEDblue, OUTPUT);  pinMode(LEDgreen, OUTPUT);  pinMode(LEDred, OUTPUT);  }  void **loop**(){  digitalWrite(LEDblue, brightnessblue); //bật blue  delay(b); //nghỉ    digitalWrite(LEDblue, dark); //tắt blue    digitalWrite(LEDred, brightnessred); //bật red  delay(b); //Break    digitalWrite(LEDred, dark); // tắt red    digitalWrite(LEDgreen, brightnessgreen); //bật green  delay(b); //Break  digitalWrite(LEDgreen, dark); //tắt green  } |

# Bài 6. Điều chỉnh độ sáng Led bằng chiết áp

## Mô tả

Bài này, thực hành lập trình arduino điều chỉnh độ sáng của đèn Led thông qua một biến trở (chiết áp). Led được đấu ởchân thứ 9, biến trở được đấu ở chân A0.

## Sơ đồ thiết kế

Hình 6. Sơ đồ bảng mạch Arduino

## Đặc điểm linh kiện

* Led: D1 (màu xanh lá canh), (10Watt)
* Điện trở: R1 (100 Ω)
* Chiếc áp
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| int x = 0;  void **setup**()  {  pinMode(A0, INPUT);  pinMode(9, OUTPUT);  }  void **loop**()  {  x = analogRead(A0); //đọc giá trị từ biến trở, cất vào x  int brightness = map(x,0,1023,0,255); //qui đổi sang độ sáng  analogWrite(9, brightness) ;//sáng led theo brightness  } |

# Bài 7. Led 7 đoạn chung cực âm

## Mô tả

Đây là bài thực hành lập trình arduino điều khiển đèn 7 đoạn chung cực âm, mỗi một giây sẽ đổi sang 1 số từ 0 đến 9 liên tục. Chân a đèn 7 đoạn được đấu ở chân 6 của arduino, chân b đấu với chân 5, chân c đấu với chân 4, chân d đấu với chân 3, chân e đấu với chân 2, chân f đấu với chân 1, và chân g đấu với chân 0.

## Sơ đồ thiết kế

Hình 7.Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Led 7 đoạn: chung cực âm, màu đỏ
* Bảng mạch điện trở: RN1 (100 Ω)
* Mạch Arduino UNO

## 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,HIGH);  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); // đợi 1 giây  Mot();  delay(1000);// đợi 1 giây  Hai();  delay(1000);// đợi 1 giây  Ba();  delay(1000);// đợi 1 giây  Bon();  delay(1000);// đợi 1 giây  Nam();  delay(1000);// đợi 1 giây  Sau();  delay(1000);// đợi 1 giây  Bay();  delay(1000);// đợi 1 giây  Tam();  delay(1000);// đợi 1 giây  Chin();  delay(1000);// đợi 1 giây  } |

# Bài 8. Led 7 đoạn chung cực âm với bộ chuyển đổi BCD

## Mô tả

Bài này, thực hành lập trình arduino điều khiển đèn 7 đoạn chung cực âm với bộ chuyển đổi BCD, mỗi một giây sẽ đổi sang 1 số từ 0 đến 9 liên tục. Chân A của Bộ chuyển BCD đấu với chân 13 của Arduino, chân B đấu với chân 12, chân C đấu với chân 11, và chân D đấu với chân 10.

## Sơ đồ thiết kế

Hình 8.Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Led 7 đoạn: chung cực âm, màu đỏ
* Bảng mạch điện trở: RN1 (100 Ω)
* Bộ chuyển đổi BCD: Thay thế tương đương 1 ký số ở hệ 10 bằng tổ hợp 4 bít tương ứng
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| int a=13, b=12, c=11, d=10;  void setup() {  pinMode(a, OUTPUT);  pinMode(b, OUTPUT);  pinMode(c, OUTPUT);  pinMode(d, OUTPUT);  }  void Khong(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,LOW);  }  void Mot(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,HIGH);  }  void Hai(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,HIGH); digitalWrite(a,LOW);  }  void Ba(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,HIGH); digitalWrite(a,HIGH);  }  void Bon(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,LOW); digitalWrite(a,LOW);  }  void Nam(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,LOW); digitalWrite(a,HIGH);  }  void Sau(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,HIGH); digitalWrite(a,LOW);  }  void Bay(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,HIGH); digitalWrite(a,HIGH);  }  void Tam(){  digitalWrite(d,HIGH); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,LOW);  }  void Chin(){  digitalWrite(d,HIGH); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,HIGH);  }  void loop() {  Khong();  delay(1000); // đợi 1 giây  Mot();  delay(1000);// đợi 1 giây  Hai();  delay(1000);// đợi 1 giây  Ba();  delay(1000);// đợi 1 giây  Bon();  delay(1000);// đợi 1 giây  Nam();  delay(1000);// đợi 1 giây  Sau();  delay(1000);// đợi 1 giây  Bay();  delay(1000);// đợi 1 giây  Tam();  delay(1000);// đợi 1 giây  Chin();  delay(1000);// đợi 1 giây  } |

# Bài 9. Sáng 8 Led tuần tự với IC 74HC595

## Mô tả

Bài này, thực hành lập trình arduino điều khiển 8 Led sáng theo một trình tự sử dụng IC 74HC595. Led sẽ sáng dần từng led một từ dưới lên trên. 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. Led được đấu vào các chân từ 7 - 15(Q7 - Q0) của IC 74HC595.

## Sơ đồ thiết kế

Hình 9.Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Led: 8 led màu đỏ, (10 Watt)
* IC 74HC595:
* Các chân từ 1 đến 7 và chân 15 là ngõ ra của IC(ứng với Q0,Q1,…,Q7)
* Chân DS (chân số 14) là ngõ vào của IC
* Chân MR (chân 10) cấp nguồn (từ 2V – 6V) – trả IC về trạng thái bạn đầu, chân OE (chân 13) nối đất – là chân khi được tích cực thì mới cho phép điều chỉnh được giá trị ngõ ra
* Chân SHCP (chân 11) là chân đưa xung clock vào IC
* Chân STCP (chân 12) là chân latch lưu trữ xung clock vào IC
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| int \_clock = 7, \_latch = 6, \_data = 5;  void **setup**() {  pinMode(\_latch, OUTPUT);  pinMode(\_clock, OUTPUT);  pinMode(\_data, OUTPUT);  }  #Hàm tính lũy thừa x^y  int **LuyThua**(int x,int y){  int luythua = 1;  for(int i = 1; i <= y;i++)  luythua = luythua \* x;  return luythua;  }  // Hàm làm sáng LED theo sô thập phân tương ứng  void **SangLedTheoDoiSo**(int x){  digitalWrite(\_latch, LOW);  shiftOut(\_data, \_clock, LSBFIRST, x);  digitalWrite(\_latch, HIGH);  delay(500);  }  void **loop**() {  for(int i = 0; i <= 7; i++){  //Sáng từng led 1 từ dưới lên trên  // Q7 = 2^0=1 | Q6 = 2^1=2 | Q5 = 2^2=4 | ... | Q0 = 2^7 = 128  SangLedTheoDoiSo(LuyThua(2,i));  }  } |

# Bài 10. Led Matrix

## Mô tả

Bài này, thực hành lập trình arduino điều khiển Led matrix sáng lên hình "mặt cười" sử dụng hai con IC 74HC595. Led sẽ sáng từ trên xuống dưới và từ trái sang phải. Chân 11(clock) của IC thứ 1 đấu với chân 1 của Arduino, chân 14(input) của IC thứ 1 đấu với chân 2 của Arduino,và chân 12(latch) của IC thứ 1 đấu với chân 0 của Arduino, IC thứ 1 này dùng để điều khiển hàng của LED MATRIX. Còn, Chân 11(clock) của IC 2 đấu với chân 9 của Arduino, chân 14(input) của IC 2 đấu với chân 10 của Arduino,và chân 12(latch) của IC thứ 2 đấu với chân 8 của Arduino, IC thứ 2 này dùng để điều khiển cột của LED MATRIX.

## Sơ đồ thiết kế

Hình 10.Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Led Matrix: 1 Led màu đỏ
* IC 74HC595: 2 con IC
* Các chân từ 1 đến 7 và chân 15 là ngõ ra của IC(ứng với Q0,Q1,…,Q7)
* Chân DS (chân số 14) là ngõ vào của IC
* Chân MR (chân 10) cấp nguồn (từ 2V – 6V) – trả IC về trạng thái bạn đầu, chân OE (chân 13) nối đất – là chân khi được tích cực thì mới cho phép điều chỉnh được giá trị ngõ ra
* Chân SHCP (chân 11) là chân đưa xung clock vào IC
* Chân STCP (chân 12) là chân latch lưu trữ xung clock vào IC
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| int \_clock\_r = 1, \_latch\_r = 0, \_data\_r = 2;  int \_clock\_c = 9, \_latch\_c = 8, \_data\_c = 10;  void **setup**() {  pinMode(\_latch\_r, OUTPUT);  pinMode(\_clock\_r, OUTPUT);  pinMode(\_data\_r, OUTPUT);  pinMode(\_latch\_c, OUTPUT);  pinMode(\_clock\_c, OUTPUT);  pinMode(\_data\_c, OUTPUT);  }  //Hàm tính lũy thừa x^y  int **LuyThua**(int x,int y){  int luythua = 1;  for(int i = 1; i <= y;i++)  luythua = luythua \* x;  return luythua;  }  // Hàm làm sáng LED theo sô thập phân tương ứng  void **SangLedTheoDoiSoCol\_atRow**(int x,int row){  digitalWrite(\_latch\_r, LOW);  shiftOut(\_data\_r, \_clock\_r, MSBFIRST, (255 - LuyThua(2,row)));  digitalWrite(\_latch\_r, HIGH);    digitalWrite(\_latch\_c, LOW);  shiftOut(\_data\_c, \_clock\_c, MSBFIRST, x);  digitalWrite(\_latch\_c, HIGH);  delay(5);  }  //Hàm vẽ mặt cười  void **matCuoi**(){  //row 0 - sáng tại cột 2 3 4 5  SangLedTheoDoiSoCol\_atRow((LuyThua(2,2)+LuyThua(2,3)+LuyThua(2,4)+LuyThua(2,5)),0);  //row 1 - sáng tại cột 1 6  SangLedTheoDoiSoCol\_atRow((LuyThua(2,1)+LuyThua(2,6)),1);  //row 2 - sáng tại cột 0 2 5 7  SangLedTheoDoiSoCol\_atRow((LuyThua(2,0)+LuyThua(2,2)+LuyThua(2,5)+LuyThua(2,7)),2);  //row 3 - sáng tại cột 0 7  SangLedTheoDoiSoCol\_atRow((LuyThua(2,0)+LuyThua(2,7)),3);  //row 4 sáng tai cột 0 1 6 7  SangLedTheoDoiSoCol\_atRow((LuyThua(2,0)+LuyThua(2,1)+LuyThua(2,6)+LuyThua(2,7)),4);  //row 5 sáng tại cột 0 2 3 4 5 7  SangLedTheoDoiSoCol\_atRow((LuyThua(2,0)+LuyThua(2,2)+LuyThua(2,3)+LuyThua(2,4)  +LuyThua(2,5)+LuyThua(2,7)),5);  //row 6 - sáng tại cột 1 6  SangLedTheoDoiSoCol\_atRow((LuyThua(2,1)+LuyThua(2,6)),6);  //row 7 - sáng tại cột 2 3 4 5  SangLedTheoDoiSoCol\_atRow((LuyThua(2,2)+LuyThua(2,3)+LuyThua(2,4)+LuyThua(2,5)),7);  }  void **loop**() {  matCuoi();  } |

# Bài 11. Cảm biến khoảng cách

## Mô tả

Bài này, thực hành lập trình arduino do khoảng cách bằng cảm biến sóng âm HC-SR04. Chân trigger của cảm biến được đấu ở chân 13 của mạch Arduino và chân Echo của cảm biến được đấu ở chân 12 của mạch Arduino.(khi chạy giả lập trên protues ta cần thêm vào cảm biến HC\_SR04 file UltraSonicTEP.HEX)

## Sơ đồ thiết kế

Hình 11.Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Cảm biến siêu âm HC-SR04: có 4 chân là: Vcc, Trig, Echo, GND.
* VCC: 5V
* Trig: Một chân Digital output
* Echo: Một chân Digital input
* GND: Đất
* Để đo khoảng cách, ta sẽ phát 1 xung rất ngắn (5 microSeconds) từ chân **Trig.**Sau đó, cảm biến sẽ tạo ra 1 xung HIGH ở chân **Echo** cho đến khi nhận lại được sóng phản xạ ở pin này. Chiều rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biển và quay trở lại.
* Distance= (travel time/2) \* speed of sound
* Biến trở (1kΩ)
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| int trig = 13; // chân trig của HC-SR04  int echo = 12; // chân echo của HC-SR04  void **setup**() {  Serial.begin(9600);  pinMode(trig, OUTPUT); // chân trig sẽ phát tín hiệu  pinMode(echo, INPUT); // chân echo sẽ thu tín hiệu  }  void **loop**() {  unsigned long thoigian;  float khoangcach;    // Phát xung từ chân trig  digitalWrite(trig,LOW); // tắt chân trig  delayMicroseconds(2);  digitalWrite(trig,HIGH); // phát xung từ chân trig  delayMicroseconds(5); // xung có độ dài 5 microSeconds    // Tính toán thời gian  //Đo độ rộng xung HIGH ở chân echo.  thoigian = pulseIn(echo, HIGH);  // Tính khoảng cách đến vật.  //Tốc độ của âm thanh trong không khí là 340 m/s  //tương đương với 0.034 cm/microSeconds ((340\*100)/10^6 )  // distance = (travel time/2)\*speed ò sound  khoangcach = (float)thoigian/2\*0.034;    // In kết quả ra Serial Monitor  Serial.print(khoangcach);  Serial.println("cm");  delay(200);  } |

# Bài 12. Điều khiển Motor DC

## Mô tả

Bài này, sẽ lập trình arduino điều khiển 2 Motor thông qua IC L293D. Hai Motor sẽ chạy về phía trước với tốc độ tối đa và sau 2 giây sẽ đổi hướng và sau 2 giây nữa sẽ ngừng, tiếp theo sau 1 giây hai motor sẽ tăng tốc từ 0 đến tốc độ tối đa, và giảm tốc từ tối đa về 0, và lập lại tất cả hoạt động trên sau 1 giây(liên tục).Hai Motor được đấu vào các chân output của L293D, Chân IN1 của L293D đấu với chân 10 của Arduino, chân IN2 của L293D đấu với chân 9 của Arduino, chân EN1 của L293D đấu với chân 8 của Arduino, và Chân IN3 của L293D đấu với chân 6 của Arduino, chân IN4 của L293D đấu với chân 5 của Arduino, chân EN2 của L293D đấu với chân 7 của Arduino

## Sơ đồ thiết kế

Hình 12. Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Motor: 2 motor DC
* IC L293D:
* VSS, VS: nơi cấp nguồn 5V
* GND: chân nối đất
* OUT1, OUT2: kết nối với DC motor 1
* OUT3, OUT4: kết nối với DC motor 2
* IN1, IN2: điều khiển hướng của DC motor 1
* Khi IN1 Low (0) và IN2 Low (0): motor ngừng
* Khi IN1 High (1) và IN2 Low (0): motor tiến về trước
* Khi IN1 Low (0) và IN2 High (1): motor lùi về sau
* Khi IN1 High (1) và IN2 High (1): motor ngừng
* IN3, IN4: điều khiển hướng của DC motor 2
* Khi IN3 Low (0) và IN4 Low (0): motor ngừng
* Khi IN3 High (1) và IN4 Low (0): motor tiến về trước
* Khi IN3 Low (0) và IN4 High (1): motor lùi về sau
* Khi IN3 High (1) và IN4 High (1): motor ngừng
* EN1: bật, tắt và điều khiển tốc độ của motor 1 (nhận giá trị từ 0 - 255)
* EN2: bật, tắt và điều khiển tốc độ của motor 2 (nhận giá trị từ 0 - 255)
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| //motor a  int enA = 8;  int in1 = 10;  int in2 = 9;  //motor b  int enB = 7;  int in3 = 6;  int in4 = 5;  void **setup**() {  pinMode(enA, OUTPUT);  pinMode(enB, OUTPUT);  pinMode(in1, OUTPUT);  pinMode(in2, OUTPUT);  pinMode(in3, OUTPUT);  pinMode(in4, OUTPUT);    //Tắt motor  digitalWrite(in1, LOW);  digitalWrite(in2, LOW);  digitalWrite(in3, LOW);  digitalWrite(in4, LOW);  }  void **loop**(){  directionControl();  delay(1000);  speedControl();  delay(1000);  }  void **directionControl**(){  analogWrite(enA, 255);  analogWrite(enB, 255);    //bật motor A và B  digitalWrite(in1, HIGH);  digitalWrite(in2, LOW);  digitalWrite(in3, HIGH);  digitalWrite(in4, LOW);  delay(2000);    //Thay đổi hướng motor  digitalWrite(in1, LOW);  digitalWrite(in2, HIGH);  digitalWrite(in3, LOW);  digitalWrite(in4, HIGH);  delay(2000);    //tắt các motor  digitalWrite(in1, LOW);  digitalWrite(in2, LOW);  digitalWrite(in3, LOW);  digitalWrite(in4, LOW);  }  void **speedControl**(){  //bật motor A và B  digitalWrite(in1, LOW);  digitalWrite(in2, HIGH);  digitalWrite(in3, LOW);  digitalWrite(in4, HIGH);  //tăng tốc độ từ 0 đến tối đa  for(int i = 0; i <= 255; i++){  analogWrite(enA, i);  analogWrite(enB, i);  delay(20);  }    //giảm tốc độ từ tối đa về 0  for(int i = 255; i >= 0; i--){  analogWrite(enA, i);  analogWrite(enB, i);  delay(20);  }    //tắt các motor  digitalWrite(in1, LOW);  digitalWrite(in2, LOW);  digitalWrite(in3, LOW);  digitalWrite(in4, LOW);  } |

# Bài 13. Hiển thị LCD (không sử dụng thư viện)

## Mô tả

Bài này, thực hành lập trình trên Arduino để màn hình LCD hiển thị mà không cần dùng thư viện bên ngoài. Chân DB0 - DB7 của LCD được đấu vào chân 2 - 9 của board mạch Arduino, chân RS và E của LCD được đấu vào chân 0 và 1 của board mạch, Chân RW của LCD được đấu vào đất để mặc định luôn luôn ở chế độ ghi.

## Sơ đồ thiết kế

Hình 13. Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Màn hình LCD:
* GND: nối đất
* VCC: nối nguồn
* V0: độ tương phản (nối vào biến trở)
* RS: lựa chọn thanh ghi (0: chọn thanh ghi lệnh; 1: chọn thanh ghi dữ liệu)
* RW: chọn thanh ghi đọc/viết dữ liệu (0: thanh ghi viết; 1: thanh ghi đọc)
* E: Enable
* DB0 – DB7: chân truyền dư liệu (8 bit: DB0DB7)
* LED Anode: cực dương led nền (0V – 5V)
* LED Cathode: cực âm led nền (0v)
* Biến trở: 1 biến trở (1 kΩ)
* Điện trở: 1 điện trở (250 Ω)
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| int D0=2,D1=3,D2=4,D3=5,D4=6,D5=7,D6=8,D7=9;  int RS=0,E=1;  void **setup**() {  pinMode(D0, OUTPUT);  pinMode(D1, OUTPUT);  pinMode(D2, OUTPUT);  pinMode(D3, OUTPUT);  pinMode(D4, OUTPUT);  pinMode(D5, OUTPUT);  pinMode(D6, OUTPUT);  pinMode(D7, OUTPUT);  pinMode(RS, OUTPUT);  pinMode(E, OUTPUT);    // Xoa noi dung tren LCD  sendCMD2LCD(0x01);  // Bat hien thi man hinh, tat con tro  sendCMD2LCD(0x0C);  // Test thu chuoi Hello  sendString2LCD("Hello K61 NTU");  // Xuong dong 2  sendCMD2LCD(0x38); // CHE DO 2 DONG  sendCMD2LCD(0xC0); // TRO XUONG DONG 2  sendString2LCD("Phan Phuc ");  char str[10];  itoa(333,str);  sendString2LCD(str);  delay(1000);  }  void **loop**() { }  // Ham gui data/cmd ra 8 chan cua LCD  void **send8BitLCD**(char D) {  // Dem D and so hoc voi 2^i de biet bit thu i =0 hay khac ko  // tim gia tri cac bit  int b0,b1,b2, b3,b4, b5, b6, b7;  if ((D & 1) == 0) b0=0; // 1== 2^0  else b0 =1;  if ((D & 2) == 0) b1=0; // 2== 2^1  else b1 =1;  if ((D & 4) == 0) b2=0; // 4== 2^2  else b2 =1;  if ((D & 8) == 0) b3=0; // 8== 2^3  else b3 =1;  if ((D & 16) == 0) b4=0; // 16== 2^4  else b4 =1;  if ((D & 32) == 0) b5=0; // 32== 2^5  else b5 =1;  if ((D & 64) == 0) b6=0; // 64== 2^6  else b6 =1;  if ((D & 128) == 0) b7=0; // 128== 2^7  else b7 =1;  digitalWrite(D0,b0);  digitalWrite(D1,b1);  digitalWrite(D2,b2);  digitalWrite(D3,b3);  digitalWrite(D4,b4);  digitalWrite(D5,b5);  digitalWrite(D6,b6);  digitalWrite(D7,b7);  }  // Ham gui lenh  void **sendCMD2LCD**(char cmd) {  //B1. Done  //B2. Dat chan RS =0, de noi rang cmd là lenh  digitalWrite(RS, LOW);  //B3. Gui 8 bit CMD vao 8 pin  send8BitLCD(cmd);  //B4. Enable cho cmd-->lcd  digitalWrite(E, LOW);  digitalWrite(E, HIGH);  //  delay(1);  }  // Ham gui ky tu hien thi  void **sendChar2LCD**(char Char) {  //B1. Done  //B2. Dat chan RS =1,  digitalWrite(RS, HIGH);  //B3. Gui 8 bit CMD vao 8 pin  send8BitLCD(Char);  //B4. Enable cho cmd-->lcd  digitalWrite(E, LOW);  digitalWrite(E, HIGH);  //  delay(1);  }  void **sendString2LCD**(char \*str) {  for (int i=0; str[i] != '\0'; i++) {  sendChar2LCD(str[i]);  }  }  // hàm chuyển số nguyên thành chuỗi  int **itoa**(int value,char \*ptr) {  int count=0,temp;  if(ptr==NULL)  return 0;  if(value==0)  {  \*ptr='0';  return 1;  }  if(value<0)  {  value\*=(-1);  \*ptr++='-';  count++;  }  for(temp=value;temp>0;temp/=10,ptr++);  \*ptr='\0';  for(temp=value;temp>0;temp/=10)  {  \*--ptr=temp%10+'0';  count++;  }  return count;  } |

# Bài 14. Bật tắt bóng đèn với cảm biến độ sáng

## Mô tả

Bài này, lập trình để thực hiện việc bật tắt bóng đèn, đèn sẽ bật khi cảm biến độ sáng thu được giá trị bé hơn giá trị đối đa của cảm biến độ sáng (0-1023). Bóng đèn được đấu vào cổng 13 của bo mạch Arduino. Cảm biến độ sáng được đấu vào cổng A0 của bo mạch Arduino.

## Sơ đồ thiết kế

Hình 14. Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Bóng đèn dây đốt: công suất 25W
* Điện trở: 2 điện trở (100 Ω)
* Mạch Arduino UNO
* Chiết áp
* Màn hình LCD
* Cảm biết độ sáng: trả ra giá trị từ (0 - 1023)

## Mã lệnh chính

|  |
| --- |
| // include thư viện để sử dụng LCD:  #include <LiquidCrystal.h>  // khởi tạo thư viện với số lượng chân giao diện  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  void setup() {  // thiết lập số hàng và cột của LD  lcd.begin(16, 2);  lcd.print("Do sang:");  pinMode(A0,INPUT);  pinMode(13, OUTPUT);  }  void loop() {  //Đọc giá trị độ sáng thu được vào biến x  // giá tri thu được từ 0 - 1023  int x = analogRead(A0);    lcd.setCursor(0, 1);  lcd.print(x); // in biến x ra LCD  delay(1000); // Đợi 1 giây    if(x < 1023/2){  digitalWrite(13, HIGH);  }  else  digitalWrite(13, LOW);  } |

# Bài 15. Điều khiển hiển thị từ 0.0 – 9.9 bằng 2 Led 7 đoạn

## Mô tả

Đây là bài thực hành lập trình arduino điều khiển hiển thị từ 0.0 đến 9.9 bằng 2 Led 7 đoạn chung cực âm thông qua 2 thiết bị 7 Segment Binary Coded Decimal BCD (IC CD4511) và kết nối lần lượt với mạch Arduino qua các cổng 0 đến cổng 4 đối với Led 1, và từ 8 đến 11 đối với Led thứ 2.

## Sơ đồ thiết kế

Hình 15. Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Led 7 đoạn: 2 Led chung cực âm, màu đỏ
* IC CD4511: 2 cái, giải mã từ số nhị phân sang số thập phân để hiển thị Led 7 đoạn.
* Điện trở: 14 điện trở (100Ω)
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| int a=0, b=1, c=2, d=3;  int a2=8, b2=9, c2=10, d2=11;  void setup() {  pinMode(a, OUTPUT);  pinMode(b, OUTPUT);  pinMode(c, OUTPUT);  pinMode(d, OUTPUT);    pinMode(a2, OUTPUT);  pinMode(b2, OUTPUT);  pinMode(c2, OUTPUT);  pinMode(d2, OUTPUT);  }  void loop() {  digitalWrite(4,HIGH);  for(int i = 0; i <= 9; i++){  if(i==0) Khong();  else if(i==1) Mot();  else if(i==2) Hai();  else if(i==3) Ba();  else if(i==4) Bon();  else if(i==5) Nam();  else if(i==6) Sau();  else if(i==7) Bay();  else if(i==8) Tam();  else if(i==9) Chin();  for(int j = 0; j <= 9; j++){  if(j==0) Khong2();  else if(j==1) Mot2();  else if(j==2) Hai2();  else if(j==3) Ba2();  else if(j==4) Bon2();  else if(j==5) Nam2();  else if(j==6) Sau2();  else if(j==7) Bay2();  else if(j==8) Tam2();  else if(j==9) Chin2();  delay(1000);  }  }  }  void Khong(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,LOW);  }  void Khong2(){  digitalWrite(d2,LOW); digitalWrite(c2,LOW);  digitalWrite(b2,LOW); digitalWrite(a2,LOW);  }  void Mot(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,HIGH);  }  void Mot2(){  digitalWrite(d2,LOW); digitalWrite(c2,LOW);  digitalWrite(b2,LOW); digitalWrite(a2,HIGH);  }  void Hai(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,HIGH); digitalWrite(a,LOW);  }  void Hai2(){  digitalWrite(d2,LOW); digitalWrite(c2,LOW);  digitalWrite(b2,HIGH); digitalWrite(a2,LOW);  }  void Ba(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,HIGH); digitalWrite(a,HIGH);  }  void Ba2(){  digitalWrite(d2,LOW); digitalWrite(c2,LOW);  digitalWrite(b2,HIGH); digitalWrite(a2,HIGH);  }  void Bon(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,LOW); digitalWrite(a,LOW);  }  void Bon2(){  digitalWrite(d2,LOW); digitalWrite(c2,HIGH);  digitalWrite(b2,LOW); digitalWrite(a2,LOW);  }  void Nam(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,LOW); digitalWrite(a,HIGH);  }  void Nam2(){  digitalWrite(d2,LOW); digitalWrite(c2,HIGH);  digitalWrite(b2,LOW); digitalWrite(a2,HIGH);  }  void Sau(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,HIGH); digitalWrite(a,LOW);  }  void Sau2(){  digitalWrite(d2,LOW); digitalWrite(c2,HIGH);  digitalWrite(b2,HIGH); digitalWrite(a2,LOW);  }  void Bay(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,HIGH); digitalWrite(a,HIGH);  }  void Bay2(){  digitalWrite(d2,LOW); digitalWrite(c2,HIGH);  digitalWrite(b2,HIGH); digitalWrite(a2,HIGH);  }  void Tam(){  digitalWrite(d,HIGH); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,LOW);  }  void Tam2(){  digitalWrite(d2,HIGH); digitalWrite(c2,LOW);  digitalWrite(b2,LOW); digitalWrite(a2,LOW);  }  void Chin(){  digitalWrite(d,HIGH); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,HIGH);  }  void Chin2(){  digitalWrite(d2,HIGH); digitalWrite(c2,LOW);  digitalWrite(b2,LOW); digitalWrite(a2,HIGH);  } |

# Bài 16. Điều khiển 3 LED bằng Remote hồng ngoại

## Mô tả

Bài này, lập trình để thực hiện việc điều khiển 3 đèn led sáng và tắt, khi nhấn phím 1 trên remote thì led đỏ sẽ sáng, và nhấn một lần nữa thì led sẽ tắt, khi nhấn phím 2 trên remote thì led vàng sẽ sáng, và nhấn một lần nữa thì led sẽ tắt, khi nhấn phím 3 trên remote thì led xanh lá sẽ sáng, và nhấn một lần nữa thì led sẽ tắt, khi nhấn phím power thì tất cả các led sẽ tắt. Led đỏ được đấu vào chân 13, Led vàng được đấu vào chân 12, và Led xanh lá được đấu vào chân 11 của Arduino. Chân out của cảm biến hồng ngoại IR sensor được đấu vào chân 2 của Arduino.

## Sơ đồ thiết kế

Hình 16. Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Led: 1 led đỏ, 1 led vàng, và 1 led xanh lá
* Điện trở: 3 điện trở R1, R2, R3 (120 Ω)
* IR remote: 1 cái điều khiển từ xa
* IR sensor: 1 cảm biến hồng ngoại để nhận tín hiệu từ remote. Sau đó truyền vào chân 2 của Arduino để xử lý.
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| // Thêm thư viện hỗ trợ IR remote  #**include** <IRremote.h>  int chanNhan = 2; // chân digital 2 dùng để đọc tín hiệu  IRrecv **boThuHongNgoai**(chanNhan); // tạo đối tượng IRrecv mới  decode\_results **maNhanDuoc**;// lưu giữ kết quả giải mã tín hiệu  // lưu các chân nối với led vào biến  int led\_red = 13;  int led\_yellow = 12;  int led\_green = 11;  //trạng thái của các LED  //mặc định là false nghĩa là tắt  boolean trangThai\_RED = false;  boolean trangThai\_YELLOW = false;  boolean trangThai\_GREEN = false;  void **setup**() {  Serial.begin(9600);// serial  boThuHongNgoai.enableIRIn(); // Bắt đầu nhận tín hiệu  pinMode(chanNhan, INPUT);  pinMode(led\_red, OUTPUT);  pinMode(led\_yellow, OUTPUT);  pinMode(led\_green, OUTPUT);  }  void **loop**() {  if (boThuHongNgoai.decode(&maNhanDuoc)) { // nếu nhận được tín hiệu  //Giải mã tín hiệu, và cắt vào biến: maNhanDuoc  //Xử lý mã nhận được  **xuLyMaNhanDuoc**();  // in ra mà hình serial, để biết được mã của phím  //Serial.println(maNhanDuoc.value);  delay(200);  **boThuHongNgoai**.resume(); // nhận giá trị tiếp theo  }  }  void **xuLyMaNhanDuoc**() {  switch(maNhanDuoc.value) {  //Bật, tắt đèn đỏ khi nhấn phím 1  case 16582903:  trangThai\_RED = !trangThai\_RED;  digitalWrite(led\_red, trangThai\_RED);  break;  //Bật, tắt đèn vàng khi nhấn phím 2  case 16615543:  trangThai\_YELLOW = !trangThai\_YELLOW;  digitalWrite(led\_yellow, trangThai\_YELLOW);  break;  //Bật, tắt đèn xanh lá khi nhấn phím 3  case 16599223:  trangThai\_GREEN = !trangThai\_GREEN;  digitalWrite(led\_green, trangThai\_GREEN);  break;  //Tắt hết đèn khi nhấn phím power  case 16580863:  trangThai\_RED = trangThai\_YELLOW = trangThai\_GREEN = false;  digitalWrite(led\_red, trangThai\_RED);  digitalWrite(led\_yellow, trangThai\_YELLOW);  digitalWrite(led\_green, trangThai\_GREEN);  break;  }  } |

# Bài 17. Keypad 4x4, hiển thị giá trị trên LCD (sử dụng thư viện Keypad.h)

## Mô tả

Bài này, thực hành lập trình trên Arduino để điều khiển keypad 4x4 (sử dụng thư viện Keypad.h). Khi nhấn vào nút nào của keypad thì nút đó sẽ hiển thị lên màn hình LCD, nếu màn hình hết chỗ để hiển thị thì làm sạch màn hình để hiển thị nút vừa nhấn. Các chân R1, R2, R3, R4, C1, C2, C3, C4 của keypad được đấu lần lược vào các chân A0, A1, 11, 10, 9, 8, 7, 6 của Arduino. Các chân RS, E, DB4, DB5, DB6, DB7 của LCD lần lược được đấu vào chân 5, 4, 3, 2, A4, A5 của Arduino.

## Sơ đồ thiết kế

Hình 17. Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Màn hình LCD:
* GND: nối đất
* VCC: nối nguồn
* V0: độ tương phản (nối vào biến trở)
* RS: lựa chọn thanh ghi (0: chọn thanh ghi lệnh; 1: chọn thanh ghi dữ liệu)
* RW: chọn thanh ghi đọc/viết dữ liệu (0: thanh ghi viết; 1: thanh ghi đọc)
* E: Enable
* DB0 – DB7: chân truyền dư liệu (8 bit: DB0DB7)
* LED Anode: cực dương led nền (0V – 5V)
* LED Cathode: cực âm led nền (0v)
* Biến trở: 1 biến trở (250 kΩ)
* Điện trở: 1 điện trở (1 kΩ)
* Keypad 4x4: Gồm có 4 hàng (R1, R2, R3, R4), và 4 cột (C1, C2, C3, C4)
* Với mỗi hàng (R1 đến R4), Chọn ra hàng Ri
* Cấp cực âm (0v) cho hàng Ri
* Nếu điện áp ở chân Cj bất kì là dương (INPUT PULLUP) => chưa nhấn
* Nếu điện áp ở chân Cj bất kì là âm (INPUT PULLUP) => đang nhấn
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| // thêm thư viện Keypad.h để có thể lập trình kedpad dễ dàng  #include <Keypad.h>  // thêm thư viện LiquidCrystal.h để hiển thị LCD dễ dàng  #include <LiquidCrystal.h>  // cấu hình LCD  LiquidCrystal lcd(5, 4, 3, 2, A4, A5);  const byte ROWS = 4; // gồm có 4 hàng  const byte COLS = 4; // gồm có 4 cột  //Định nghĩa các giá trị trả về  char keys[ROWS][COLS] = {  {'1','2','3','A'},  {'4','5','6','B'},  {'7','8','9','C'},  {'\*','0','#','D'}  };  //kết nối với chân hàng của keypad (hàng 1,2,3,4)  byte rowPins[ROWS] = {A0, A1, 11, 10};  //kết nối với chân cột của keypad (cột 1,2,3,4)  byte colPins[COLS] = {9, 8, 7, 6};  int LCDCol = 0;//cột 0 của LCD  int LCDRow = 0;//hàng 0 của LCD  //cấu hình thư viện keypad  Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );  void setup(){  // thiết lập hiển thị trên LCD  Serial.begin(9600);  lcd.begin(16, 2);  lcd.setCursor(LCDCol, LCDRow);  }    void loop(){  char key = keypad.getKey();    if (key){  //xuất lên LCD để xem kết quả  Serial.println(key);    if ( LCDCol > 15 ){  // nếu cột đã lớn hơn 15 rồi thì xuống hàng mới (hết hàng 0)  ++LCDRow;    // nếu LCD đã hết chỗ để hiển thị thì làm sạch nó đi  // rồi hiển thị số vừa nhấn  if (LCDRow>1)  { LCDRow=0; LCDCol = 0 ; lcd.clear(); }  //khi xuống hàng mới thì thiết lập lại con trỏ ở cột 0  //(hàng 1 cột 0)  LCDCol = 0 ;  }    // thiệt lập con trỏ hiển thị của LCD  lcd.setCursor (LCDCol, LCDRow);  lcd.print(key);  // khi hiển thị 1 giá trị thì tăng cột lên để  //chuẩn bị hiển thị giá trị tiếp theo  ++LCDCol;  }  } |

# Bài 18. Điều chỉnh đèn và quạt theo độ sáng và nhiệt độ môi trường.

## Mô tả

Bài này, thực hành lập trình trên Arduino để điều khiển độ sáng của bóng đèn và tốc độ quay của quạt (DC motor). Nhiệt độ và độ sáng của môi trường sẽ được hiển thị trên màn hình LCD (sử dụng thư viện LiquidCrystal.h). Bóng đèn sẽ sáng tương ứng với độ sáng của môi trường, khi độ sáng của môi trường lớn thì bóng đèn sẽ sáng yếu hoặc tắt luôn và ngược lại, khi độ sáng môi trường nhỏ thì bóng đèn sẽ sáng mạnh tương ứng. Đối với quạt thì sẽ có 4 mức quay, khi nhiệt độ lớn hơn 32 C quạt sẽ quay mạnh , khi nhiệt độ từ 26 đến 32 C sẽ quay vừa, khi nhiệt độ từ 24 đến 26 độ sẽ quay yếu, và khi nhiệt độ bé hơn 24 C sẽ tắt quạt. Các chân RS, E, DB4, DB5, DB6, DB7 của LCD lần lược được đấu vào chân 12, 11, 5, 4, 3, 2 của Arduino. Bóng đèn được đấu vào chân 10, động cơ DC được đấu vào chân 9 của Arduino. Cảm biến độ sáng được đấu vào chân A0, và cảm biến nhiệt độ TMP36 được đấu vào chân A1 của Arduino.

## Sơ đồ thiết kế

Hình 18. Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Màn hình LCD:
* GND: nối đất
* VCC: nối nguồn
* V0: độ tương phản (nối vào biến trở)
* RS: lựa chọn thanh ghi (0: chọn thanh ghi lệnh; 1: chọn thanh ghi dữ liệu)
* RW: chọn thanh ghi đọc/viết dữ liệu (0: thanh ghi viết; 1: thanh ghi đọc)
* E: Enable
* DB0 – DB7: chân truyền dư liệu (8 bit: DB0DB7)
* LED Anode: cực dương led nền (0V – 5V)
* LED Cathode: cực âm led nền (0v)
* Biến trở: 1 biến trở (250 kΩ)
* Điện trở: 3 điện trở (120 Ω, 10 kΩ, 220 Ω)
* Bóng đèn
* Động cơ DC
* Cảm biến độ sáng: trả ra giá trị từ (0 - 1023)
* Cảm biến nhiệt độ TMP36: có thể đo được nhiệt độ từ **-40 =>125** độ C. Khi Arduino làm việc nó sẽ trả tín hiệu tương tự này về giá trị từ 0-1023.
* Mạch Arduino UNO.

## Mã lệnh chính

|  |
| --- |
| // Thư viện để hiển thị màn hình LCD  #include <LiquidCrystal.h>  //Khởi tao thư viện để hiển thị LCD với các chân  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  // Thiết lập một số giá trị ban đầu  **int** dosang;  **int** nhietdo;  void **setup**() {  //Thiết lập số hàng và cột của LCD  lcd.begin(16, 2);  lcd.print("Do sang:");  lcd.setCursor(0, 1);  lcd.print("Nhiet do:");  pinMode(A0,INPUT);  pinMode(10, OUTPUT);  pinMode(A1, INPUT);  pinMode(9, OUTPUT);  }  void **loop**() {    //Đọc giá trị độ sáng thu được vào biến dosang  // giá tri thu được từ 0 - 1023  dosang = **analogRead**(A0);  //Thiết lập con trỏ LCD ở hàng 0 cột 10  lcd.setCursor(10, 0);  lcd.print(dosang);// in biến dosang ra LC    //Điều chỉnh độ sáng của đèn theo độ sáng của môi trường  //Chuyển đổi giá trị thu được từ cản biến độ sáng  //để có thể điều chỉnh độ sáng của bóng đèn qua arduino  **int** xulyDS = **map**(dosang,0,1023,0,255);  //thiệt lập độ sáng cho bóng đèn  **analogWrite**(10,255 - xulyDS);    //--------------------------------------------------  // Chuyển đổi giá trị đọc được từ chân A1 thành  //nhiệt độ tương ứng  nhietdo = **map**(((**analogRead**(A1) - 20) \* 3.04), 0, 1023, -40, 125);  //Thiết lập con trỏ LCD ở hàng 1 cột 10  lcd.setCursor(10, 1);  lcd.print(nhietdo);// in biến dosang ra LCD  lcd.setCursor(13, 1);  lcd.print(" C");    //xử lý tốc độ quay của máy quạt(DC motor) dựa vào nhiệt độ  //gồm có 4 mức tốc độ  if(nhietdo >= 32)  analogWrite(9, 255);  else if(nhietdo > 26 && nhietdo < 32 )  analogWrite(9, 155);  else if(nhietdo >= 24 && nhietdo <= 26)  analogWrite(9, 55);  else if(nhietdo < 24)  analogWrite(9, 0);    **delay**(100);// Đợi 100 mili giây  } |

# Bài 19. Điều khiển Led RGB bằng Remote hồng ngoại

## Mô tả

Bài này, lập trình để thực hiện việc điều khiển đèn led RGB, khi nhấn từ phím 0 đến phím 6 thì led RGD sẽ hiện sáng các màu khác nhau, và khi nhấn nút power thì led RGB sẽ tắt. Chân Red, chân Blue, chân Green của led RGB lần lượt được đấu vào các chân 11, 10, 9 của Arduino. Chân Out của cảm biến hồng ngoại sẽ chuyền tín hiệu nhận được từ Remote vào chân 2 của Arduino.

## Sơ đồ thiết kế

Hình 19. Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Led RGB: có 4 chân – chân red, blue, green và một chân nối đất
* Điện trở: 3 điện trở R1, R2, R3 (1 kΩ)
* IR remote: 1 cái điều khiển từ xa
* IR sensor: 1 cảm biến hồng ngoại để nhận tín hiệu từ remote. Sau đó truyền vào chân 2 của Arduino để xử lý.
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| // Thêm thư viện hỗ trợ IR remote  #include <IRremote.h>  **int** chanNhan = 2; // chân digital 2 dùng để đọc tín hiệu  **IRrecv** boThuHongNgoai(chanNhan); // tạo đối tượng IRrecv mới  **decode**\_results maNhanDuoc;// lưu giữ kết quả giải mã tín hiệu  // lưu các chân nối với led RGB vào biến  **int** redPIN = 11;  **int** bluePIN = 10;  **int** greenPIN = 9;  void **setup**() {  Serial.begin(9600);// serial  boThuHongNgoai.enableIRIn(); // Bắt đầu nhận tín hiệu  pinMode(chanNhan, INPUT);  pinMode(redPIN, OUTPUT);  pinMode(bluePIN, OUTPUT);  pinMode(greenPIN, OUTPUT);  }  void **loop**() {  if (boThuHongNgoai.decode(&maNhanDuoc)) { // nếu nhận được tín hiệu  //Giải mã tín hiệu, và cắt vào biến: maNhanDuoc  //Xử lý mã nhận được  **xuLyMaNhanDuoc**();  // in ra mà hình serial, để biết được mã của phím  //Serial.println(maNhanDuoc.value);  **delay**(40);  boThuHongNgoai.**resume**(); // nhận giá trị tiếp theo  }  }  void **xuLyMaNhanDuoc**() {  switch(maNhanDuoc.value) {  //nhấn phím Power để tắt LED  case 16580863:  analogWrite(redPIN,0);  analogWrite(greenPIN,0);  analogWrite(bluePIN,0);  break;  //nhấn phím 0 để hiện màu Cam  case 16593103:  analogWrite(redPIN,255);  analogWrite(greenPIN,128);  analogWrite(bluePIN,0);  break;  //nhấn phím 1 để hiện màu Đỏ  case 16582903:  analogWrite(redPIN,255);  analogWrite(greenPIN,0);  analogWrite(bluePIN,0);  break;  //nhấn phím 2 để hiện màu Xanh lá chanh  case 16615543:  analogWrite(redPIN,0);  analogWrite(greenPIN,255);  analogWrite(bluePIN,0);  break;  //nhấn phím 3 để hiện màu Xanh Dương  case 16599223:  analogWrite(redPIN,0);  analogWrite(greenPIN,0);  analogWrite(bluePIN,255);  break;  //nhấn phím 4 để hiện màu Vàng  case 16591063:  analogWrite(redPIN,255);  analogWrite(greenPIN,255);  analogWrite(bluePIN,0);  break;  //nhấn phím 5 để hiện màu Aqua  case 16623703:  analogWrite(redPIN,0);  analogWrite(greenPIN,255);  analogWrite(bluePIN,255);  break;  //nhấn phím 6 để hiện màu Hồng Tím  case 16607383:  analogWrite(redPIN,255);  analogWrite(greenPIN,0);  analogWrite(bluePIN,255);  break;  }  } |

# Bài 20. Cảm biến nhiệt độ - độ ẩm DHT11 (sử dụng thư viện DHT.h)

## Mô tả

Bài này, thực hiện lập trình Arduino với cảm biến nhiệt độ, độ ẩm DHT11. Nhiệt độ và độ ẩm thu được từ cảm biến sẽ hiển thị trên màn hình LCD. Chân DATA của cảm biến DHT11 sẽ chuyền dữ liệu vào chân 12 của Arduino. Chân D0 - D7 của LCD sẽ được đấu vào chân 2 đến chân 9 của Arduino, chân RS và chân E của LCD được đấu vào chân 0 và 1 của Arduino, chân VEE của LCD được đấu vào biến trở, và chân RW của LCD được đấu vào đất để mặc định luôn luôn ở chế độ ghi.

## Sơ đồ thiết kế

Hình 20. Sơ đồ mạch Arduino.

## Đặc điểm linh kiện

* Màn hình LCD:
* GND: nối đất
* VCC: nối nguồn
* V0: độ tương phản (nối vào biến trở)
* RS: lựa chọn thanh ghi (0: chọn thanh ghi lệnh; 1: chọn thanh ghi dữ liệu)
* RW: chọn thanh ghi đọc/viết dữ liệu (0: thanh ghi viết; 1: thanh ghi đọc)
* E: Enable
* DB0 – DB7: chân truyền dư liệu (8 bit: DB0DB7)
* Biến trở: 1 biến trở (1 kΩ)
* Cảm biến nhiệt độ, độ ẩm DHT11:
* Điện áp hoạt động: 3V – 5V(DC)
* Dãi độ ẩm hoạt động: 20% - 90% RH, sai số ±5%RH
* Dãi nhiệt độ hoạt động: 0°C ~ 50°C, sai số ±2°C
* Khoảng cách truyển tối đa: 20m
* Có 3 chân: VCC, GND, DATA(để chuyền dữ liệu đến Arduino)
* Mạch Arduino UNO

## Mã lệnh chính

|  |
| --- |
| //Thêm thư viện DHT.h để sử dụng cảm biến DHT11 dễ dàng  #include "DHT.h"  //Khai báo chân kết nối cho cảm biến DHT ở đây mình dùng chân D4 trên Arduino Uno.  **int** DHTPIN = 12;  **int** DHTTYPE = DHT11;  **int** D0=2,D1=3,D2=4,D3=5,D4=6,D5=7,D6=8,D7=9;  **int** RS=0,E=1;  //Thiết lập cho thư viện DHT.h  **DHT** dht(DHTPIN, DHTTYPE);  void **setup**()  {  dht.begin(); //Khởi động thư viện  pinMode(D0, OUTPUT);  pinMode(D1, OUTPUT);  pinMode(D2, OUTPUT);  pinMode(D3, OUTPUT);  pinMode(D4, OUTPUT);  pinMode(D5, OUTPUT);  pinMode(D6, OUTPUT);  pinMode(D7, OUTPUT);  pinMode(RS, OUTPUT);  pinMode(E, OUTPUT);  }  void **loop**()  {  **float** h = dht.readHumidity(); // Đọc giá trị nhiệt độ từ cảm biến  **float** t = dht.readTemperature(); // Đọc giá trị độ ẩm từ cảm biến  // Xoa noi dung tren LCD  **sendCMD2LCD**(0x01);  // Bat hien thi man hinh, tat con tro  **sendCMD2LCD**(0x0C);    //In nhiet đô ra màn hình LCD  **sendString2LCD**("Nhiet do: ");  **String** nhietdo = String(t, 1); // chuyển float thành chuỗi sử dụng float và các vị trí thập phân  **sendString2LCD**(nhietdo);  // Xuong dong 2  **sendCMD2LCD**(0x38); // CHE DO 2 DONG  **sendCMD2LCD**(0xC0); // TRO XUONG DONG 2  //In độ ẩm ra màn hình LCD  **sendString2LCD**("Do am: ");  **String** doam = String(h, 1); // chuyển float thành chuỗi sử dụng float và các vị trí thập phân  **sendString2LCD**(doam);  **delay**(100);  }  // Ham gui data/cmd ra 8 chan cua LCD  void **send8BitLCD**(char D) {  // Dem D and so hoc voi 2^i de biet bit thu i =0 hay khac ko  // tim gia tri cac bit  int b0,b1,b2, b3,b4, b5, b6, b7;  if ((D & 1) == 0) b0=0; // 1== 2^0  else b0 =1;  if ((D & 2) == 0) b1=0; // 2== 2^1  else b1 =1;  if ((D & 4) == 0) b2=0; // 4== 2^2  else b2 =1;  if ((D & 8) == 0) b3=0; // 8== 2^3  else b3 =1;  if ((D & 16) == 0) b4=0; // 16== 2^4  else b4 =1;  if ((D & 32) == 0) b5=0; // 32== 2^5  else b5 =1;  if ((D & 64) == 0) b6=0; // 64== 2^6  else b6 =1;  if ((D & 128) == 0) b7=0; // 128== 2^7  else b7 =1;  digitalWrite(D0,b0);  digitalWrite(D1,b1);  digitalWrite(D2,b2);  digitalWrite(D3,b3);  digitalWrite(D4,b4);  digitalWrite(D5,b5);  digitalWrite(D6,b6);  digitalWrite(D7,b7);  }  // Ham gui lenh  void **sendCMD2LCD**(char cmd) {  //B1. Done  //B2. Dat chan RS =0, de noi rang cmd là lenh  digitalWrite(RS, LOW);  //B3. Gui 8 bit CMD vao 8 pin  send8BitLCD(cmd);  //B4. Enable cho cmd-->lcd  digitalWrite(E, LOW);  digitalWrite(E, HIGH);  //  delay(1);  }  // Ham gui ky tu hien thi  void **sendChar2LCD**(char Char) {  //B1. Done  //B2. Dat chan RS =1,  digitalWrite(RS, HIGH);  //B3. Gui 8 bit CMD vao 8 pin  send8BitLCD(Char);  //B4. Enable cho cmd-->lcd  digitalWrite(E, LOW);  digitalWrite(E, HIGH);  //  delay(1);  }  void **sendString2LCD**(String str) {  for (int i=0; str[i] != '\0'; i++) {  sendChar2LCD(str[i]);  }  } |

Chủ đề 2: Lập trình với STM32

# Bài 1. Nháy Led -STM32

## Mô tả

Đây là bài thực hành lập trình trên board mạch STM32, trong bài này sẽ làm cho 4 đèn Led nhấp nháy liên tục với thời gian delay là 1 giây. Bốn đèn sẽ lần lượt được đấu vào chân A12, PA13, PA14, PA15 của STM32.

## Sơ đồ thiết kế

Hình 1. Sơ đồ mạch STM32F401VE.

## Đặc điểm linh kiện

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

## 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 2. Nháy Led Trái Tim - STM32

## Mô tả

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

## Sơ đồ thiết kế

Hình 2. Sơ đồ mạch STM32F401VE.

## Đặc điểm linh kiện

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

## Mã lệnh chính

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

# Bài 3. Nút Bấm - STM32

## Mô tả

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

## Sơ đồ thiết kế

Hình 3. Sơ đồ mạch STM32F401VE.

## Đặc điểm linh kiện

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

## Mã lệnh chính

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

# Bài 4. Lập Trình Ngắt - STM32

## Mô tả

Đây là bài thực hành lập trình trên board mạch STM32, trong bài này sẽ thực hiện lập trình ngắt. Tiến trình chính sẽ liên tục nhấp nháy 4 Led. Còn trong tiến trình phụ, khi nhấn và thả nút bấm thì sẽ làm cho 1 Led và Motor hoạt động, và nhấn thả nút thêm lần nữa thì sẽ làm Led và Motor đó ngừng hoạt động. 4 Led của tiến trình chính được đấu vào chân PD0 - PD3 của STM32, ở tiến trình 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 4. Sơ đồ mạch STM32F401VE.

## Đặc điểm linh kiện

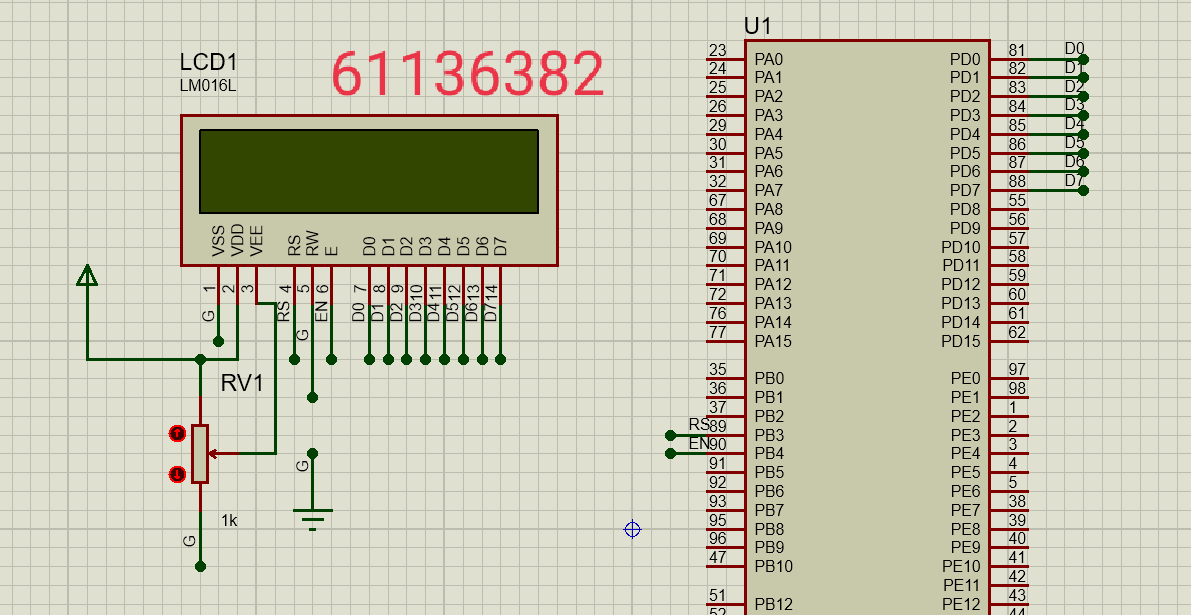
* Led: 5 LED (màu xanh)
* Điện trở: 6 điện trở (10k Ω ,5 điện trở 150 Ω)
* Motor: 1 động cơ 1 chiều
* Nút bấm
* Mạch STM32F401VE

## Mã lệnh chính

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

# Bài 5. Màn hình LCD - STM32

## Mô tả

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

## Sơ đồ thiết kế

Hình 5. Sơ đồ mạch STM32F401VE.

## Đặc điểm linh kiện

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

## Mã lệnh chính

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

# Bài 6. Điều kiển Motor - STM32 (lập trình timer cơ bản)

## Mô tả

Đây là bài thực hành lập trình trên board mạch STM32 điều khiển 2 Motor thông qua IC L239D. Hai Motor sẽ chạy về phía trước với tốc độ tối đa và sau 2 giây sẽ đổi hướng và sau 2 giây nữa sẽ ngừng, tiếp theo sau 1 giây hai motor sẽ tăng tốc từ 0 đến tốc độ tối đa, và giảm tốc từ tối đa về 0, và lập lại tất cả hoạt động trên sau 1 giây(liên tục). Hai Motor được đấu vào các chân output của L239D, Chân IN1 của L239D đấu với chân PD10 của STM32, chân IN2 của L239D đấu với chân PD11 của STM32, chân EN1 của L239D đấu với chân PD12 của STM32, và Chân IN3 của L239D đấu với chân PD14 của STM32, chân IN4 của L239D đấu với chân PD15 của STM32, chân EN2 của L239D đấu với chân PD13 của STM32.

## Sơ đồ thiết kế

Hình 6. Sơ đồ mạch STM32F401VE.

## Đặc điểm linh kiện

* Motor: 2 motor DC
* IC L293D:
* VSS, VS: nơi cấp nguồn 5V
* GND: chân nối đất
* OUT1, OUT2: kết nối với DC motor 1
* OUT3, OUT4: kết nối với DC motor 2
* IN1, IN2: điều khiển hướng của DC motor 1
* Khi IN1 Low (0) và IN2 Low (0): motor ngừng
* Khi IN1 High (1) và IN2 Low (0): motor tiến về trước
* Khi IN1 Low (0) và IN2 High (1): motor lùi về sau
* Khi IN1 High (1) và IN2 High (1): motor ngừng
* IN3, IN4: điều khiển hướng của DC motor 2
* Khi IN3 Low (0) và IN4 Low (0): motor ngừng
* Khi IN3 High (1) và IN4 Low (0): motor tiến về trước
* Khi IN3 Low (0) và IN4 High (1): motor lùi về sau
* Khi IN3 High (1) và IN4 High (1): motor ngừng
* EN1: bật, tắt và điều khiển tốc độ của motor 1 (nhận giá trị từ 0 - 255)
* EN2: bật, tắt và điều khiển tốc độ của motor 2 (nhận giá trị từ 0 - 255)
* Mạch STM32F401VE

## Mã lệnh chính

|  |
| --- |
| #include "main.h"  //Co san mot handle cua Timer4 la htim4  TIM\_HandleTypeDef htim4;  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  static void MX\_TIM4\_Init(void);  void directionControl();  void speedControl();  int main(void) {  HAL\_Init();  SystemClock\_Config();    MX\_GPIO\_Init();  MX\_TIM4\_Init();    //Khoi dong Timer tren cac kenh  HAL\_TIM\_PWM\_Start(&htim4, TIM\_CHANNEL\_1);  HAL\_TIM\_PWM\_Start(&htim4, TIM\_CHANNEL\_2);    while (1) {  directionControl();  HAL\_Delay(1000);  speedControl();  HAL\_Delay(1000);  }  }  //Ham dieu khien huong cua motor  void directionControl(){  //toc do toi da  \_\_HAL\_TIM\_SET\_COMPARE(&htim4,TIM\_CHANNEL\_1,255);  \_\_HAL\_TIM\_SET\_COMPARE(&htim4,TIM\_CHANNEL\_2,255);    //bat motor 1 va 2  //motor 1  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_RESET);  //motor 2  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_RESET);  HAL\_Delay(2000);    //thay doi huong motor  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_SET);  HAL\_Delay(2000);    //tat cac motor  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_RESET);  }  //Ham dieu khien toc do cua motor  void speedControl(){  //bat motor 1 va 2  //motor 1  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_RESET);  //motor 2  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_RESET);    //tang toc do tu 0 den toi da  for(int pwm = 0; pwm <= 255; pwm++){  \_\_HAL\_TIM\_SET\_COMPARE(&htim4,TIM\_CHANNEL\_1,pwm);  \_\_HAL\_TIM\_SET\_COMPARE(&htim4,TIM\_CHANNEL\_2,pwm);  HAL\_Delay(20);  }    //giam toc do tu doi da den 0  for(int pwm = 255; pwm >= 0; pwm--){  \_\_HAL\_TIM\_SET\_COMPARE(&htim4,TIM\_CHANNEL\_1,pwm);  \_\_HAL\_TIM\_SET\_COMPARE(&htim4,TIM\_CHANNEL\_2,pwm);  HAL\_Delay(20);  }    //tat cac motor  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_10,GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_11,GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_14,GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_15,GPIO\_PIN\_RESET);  } |

# Bài 7. Chuyển đổi tín hiệu analog nhận được từ chiết áp sang tín hiệu số (ADC)

## Mô tả

Đây là bài thực hành lập trình board mạch STM32, bài này sẽ điều khiển động cơ DC thông qua biến trở(chiết áp), khi điều chỉnh biến trở và board mạch thu được giá trị(0 - 4095) từ 2049 trở lên thì động cơ sẽ quay, ngược lại sẽ không quay. Biến trở được đấu vào chân PA0 và động cơ DC sẽ được đấu vào chân PD0 của STM32.

## Sơ đồ thiết kế

Hình 7. Sơ đồ mạch STM32F401VE.

## Đặc điểm linh kiện

* Motor: 1 motor DC
* Biến trở: 1 biến trở RV1 (1kΩ)
* Mạch STM32F401VE

## Mã lệnh chính

|  |
| --- |
| #include "main.h"  /\* Private variables ---------------------------------------------------------\*/  ADC\_HandleTypeDef hadc1;  DMA\_HandleTypeDef hdma\_adc1;  /\* Private function prototypes -----------------------------------------------\*/  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  static void MX\_DMA\_Init(void);  static void MX\_ADC1\_Init(void);  volatile uint32\_t res\_value;  int **main**(void) {  HAL\_Init();  SystemClock\_Config();    MX\_GPIO\_Init();  MX\_DMA\_Init();  MX\_ADC1\_Init();    **while** (1) {  HAL\_ADC\_Start\_DMA(&hadc1,(uint32\_t\*)res\_value,1);  HAL\_Delay(1000);  HAL\_ADC\_Stop\_DMA(&hadc1);  if(res\_value > 2049)  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,GPIO\_PIN\_SET);  else  HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,GPIO\_PIN\_RESET);    }  } |