Biên soạn: Nguyễn Thành Lãnh – LTN ( lớp 61-CNTT\_1)

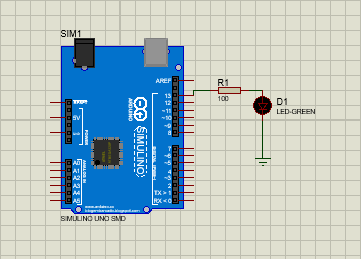
1. ARDUINO

# Bài 1. Led nhấp nháy:

## Mô tả

Bài này thực hiện việc lập trình mô phỏng mạch Arduino điều khiển một đèn led nhấp nháy trên cổng số 13 trong khoảng thời gian 1 giây.

## Sơ đồ



Hình . Mach Arduino - Nháy led 1s

## Linh kiện

* Một mạch Arduino Uno
* Một điện trở 100Ω
* Một đèn led

## Code chương trình

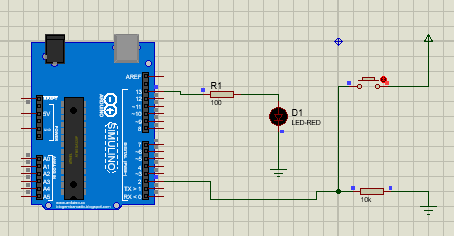
|  |
| --- |
| void setup() {  pinMode(13, OUTPUT); // OUTPUT format is port number 13  }  void loop() {  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  delay(1000); // wait for a second  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  delay(1000); // wait for a second  } |

# Bài 2. Bật tắt đèn Led bằng nút bấm

## Mô tả

Bài này thực hiện việc lập trình mô phỏng mạch Arduino điều khiển quá trình bật, tắt đèn thông qua một nút bấm.

## Sơ đồ



Hình . Mạch Arduino - Điều khiển led qua nút bấm.

## Linh kiện

* Một mạch Arduino Uno
* Một nút bấm
* Một đèn led
* Hai điện trở 100Ω và 10k Ω

## Code chương trình

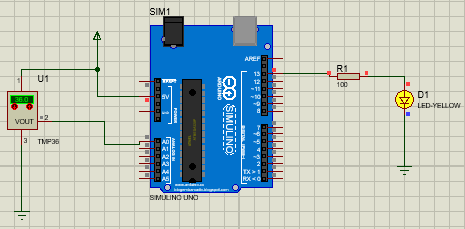
|  |
| --- |
| int x = 0;  void setup()  {  pinMode(2, INPUT); // INPUT format is port 2  pinMode(13,OUTPUT); //OUTPUT format is port 13  }  void loop()  {  x = digitalRead(2); // doc cong 2 gan cho bien x  if( x == HIGH) //Kiem tra nut co dang  {  digitalWrite(13, HIGH); //Bat den.  }  else  {  digitalWrite(13, LOW); //Tat den.  }    delay(10); // Delay a little bit to improve simulation performance  } |

# Bài 3: Làm việc với cảm biến nhiệt độ.

## Mô tả

Bài này lập trình mô phỏng mạch Arduino điều khiển độ sáng của đèn led thông quá cảm biến nhiệt độ TMP.

## Sơ đồ



Hình .Mạch Arduino - Điều khiển led qua TMP

## Linh kiện

* Một mạch Arduino Uno
* 1 đèn led
* Môt cảm biến nhiệt độ TMP
* 1 điện trở 100Ω

## Code chương trình

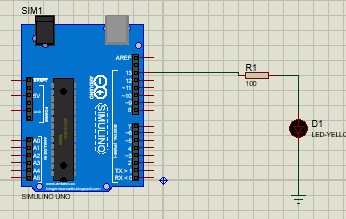
|  |
| --- |
| int a = 13; // a lay gia tri cong  void setup(){  pinMode(a,OUTPUT); // format Cong a 🡺 OUTPUT  Serial.begin(9600);  }  void loop()  {  int x = analogRead(A0); //doc A0 gan cho x  int t = map(x, 0,410,-50, 150); // tinh nhiet do tu gia tri x  if(t >= 36) // Sang den khi nhiet do lon hon hoac bang 36  digitalWrite(a,HIGH);  else  digitalWrite(a,LOW);  delay(100);  } |

# 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 Led sáng dần, rồi tắt dần

## Sơ đồ



Hình . Mạch Arduino - Điều khiển led sáng( tắt) dần

## Linh kiện

* Mạch Arduino
* Đèn led
* Điện trở 100Ω

## Code chương trình

|  |
| --- |
| int brightness = 0;  void setup(){  pinMode(13,OUTPUT);  }  void loop(){  for( brightness = 0; brightness <= 255; brightness +=5){ // sáng dần  analogWrite(13, brightness);  delay(30);  }  for(brightness = 255; brightness >= 0; brightness -= 5){ // tắt dần  analogWrite(13, brightness);  delay(30);  }  } |

# Bài 5: Điều khiển độ sáng thông qua chiết áp

## Mô tả

Bài này thực hiện lập trình, mô phỏng mạch Arduino điều khiển độ sáng của đèn thông qua chiết áp, gắn ở chân A0

## Sơ đồ

## 

Hình . Mach Arduino - Điều khiển độ sáng đèn led qua chân A0

## Linh kiện

* Một mạch Arduino
* Một chiết áp
* Một đèn
* Một điện trở 100Ω

## Code chương trình

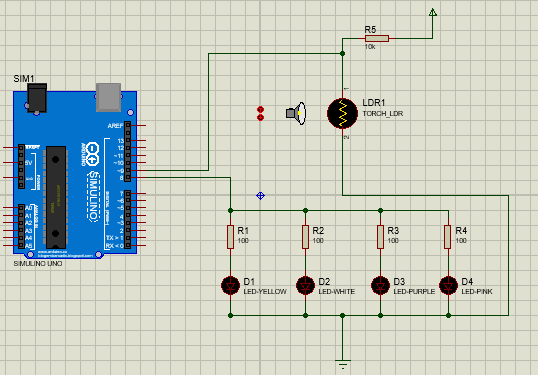
|  |
| --- |
| int X = 0;  void setup() {  pinMode(A0, INPUT);  pinMode(9, OUTPUT);  }  void loop() {  X = analogRead(A0);  int brightness = map(X,0,1023,0,255);  analogWrite(9, brightness);  } |

# Bài 6: Điều khiển đèn bằng LDR

## Mô tả

Bài này thực hiện lập trình, mô phỏng Arduino điều khiển việc bật tắt đèn thông qua cảm biến ánh sáng TORCH\_LDR

## Sơ đồ



Hình . Mạch Arduino - Điều khiểu đèn thông qua LDR

## Linh kiện

* Mạch Arduino Uno
* Bốn đèn led khác nhau
* Bốn điện trở 220Ω và một điện trở 10kΩ
* Một cảm biến ánh sáng Torch\_LDR

## Code chương trình

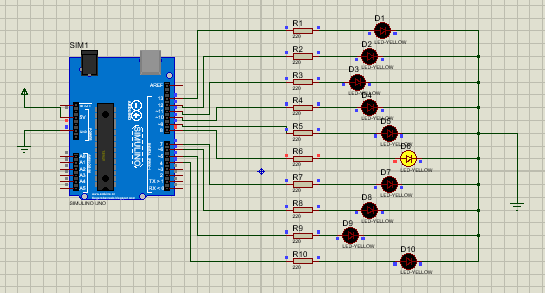
|  |
| --- |
| int LDR = 9;  int led = 8;  void setup()  {  pinMode( LDR, INPUT\_PULLUP);  pinMode( led, OUTPUT);  }  void loop()  {  if(digitalRead(ldr) == HIGH)  {  digitalWrite(led, LOW);  }  else  {  digitalWrite(led, HIGH);  }  } |

# Bài 7: LED sáng dần từ đèn 1 tới đèn 10 và ngược lại

## Mô tả

Bài này thực hiện lập trình và mô phỏng mạch Arduino điều khiển độ sáng của các đèn, sáng dần từ 1 đến 10.

## Sơ đồ



Hình . Mạch Arduino - Điều khiểu 10 led sáng dần

## Linh kiện

* Mạch Arduino
* Mười đèn led
* Mười điện trở 220Ω

## Code chương trình

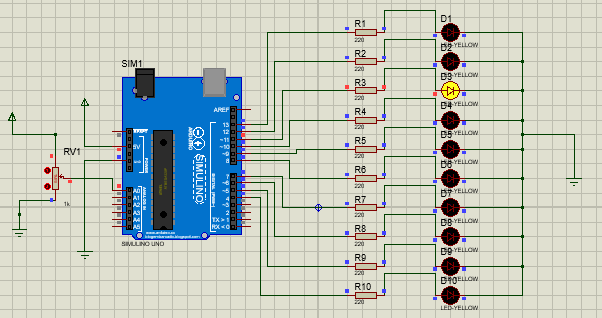
|  |
| --- |
| byte LedPin [] = {4,5,6,7,8,9,10,11,12,13}; //Mảng các cổng kết nối đèn  int direction = 1;  int currentLED = 0;  void setup(){  for (int x = 0; x < 10; x++){  pinMode(LedPin[x],OUTPUT); // lần lượt format đầu ra của đèn  }  }  void loop(){  for(int x = 0; x < 10; x++){  digitalWrite( LedPin[x],LOW); //tắt đèn  }  digitalWrite(LedPin[currentLED],HIGH);  currentLED += direction;  if (currentLED ==9) // chạy xong từ Đ1 đến Đ10 thì đổi chiều  direction = -1;  if (currentLED ==0) // chạy xong từ Đ10 đến Đ1 thì đổi chiều  direction = 1;  delay(500);  } |

# Bài 8: LED sáng dần từ 1 đến 10 và ngược lại, có sử dụng biến trở.

## Mô tả

Tương tự như bài số 7, bài này chỉ thêm vào một biến trở POT-HG nối với chân A0 của mạch Arduino, để điều chỉnh thời gian delay của đèn.

## Sơ đồ



Hình 8. Mạch Arduino - điều khiển sáng đèn có biến trở

## Linh kiện

* Một mạch Arduino
* Một biến trở POT-HG
* Mười điện trở 220Ω
* Mười đèn LED

## Code chương trình

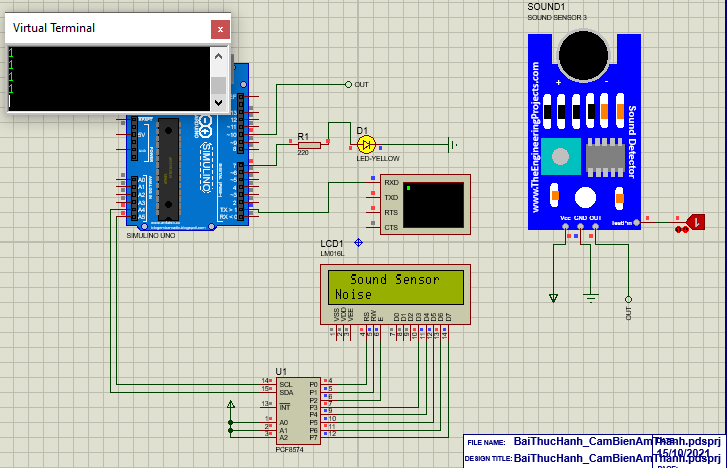
|  |
| --- |
| int ledPin[] = {4,5,6,7,8,9,10,11,12,13};  int direction = 1;  int currentLED = 0;  int potPin = 0;  unsigned long changeTime;  void setup()  {  for (int x =0; x<10;x++){  pinMode(ledPin[x], OUTPUT);  }  }  void loop(){  int delayvalu = analogRead(potPin);  for( int x = 0; x<10;x++){  digitalWrite(ledPin[x],LOW);  }  digitalWrite(ledPin[currentLED],HIGH);  currentLED += direction;  if(currentLED==9)  {  direction = -1;  }  if (currentLED == 0)  {  direction = 0;  }  delay(delayvalu);  } |

# Bài 9: Cảm biến âm thanh

## Mô tả

Bài này thực hiện lập trình, mô phỏng mạch Arduino sử dụng cảm biến âm thành để điều khiển bật tắt đèn led.

## Sơ đồ



Hình . Mạch Arduino - Mô phỏng bộ cảm biến âm thanh

## Linh kiện

* Một mạch Arduino
* Một cảm biến âm thanh Sound Sensor 3
* Một mạch mở rộng I2C PCF8574
* Điện trở 220Ω và đèn led
* Thiết bị hiển thị LM016L
* Virtural Terminal

## Code chương trình

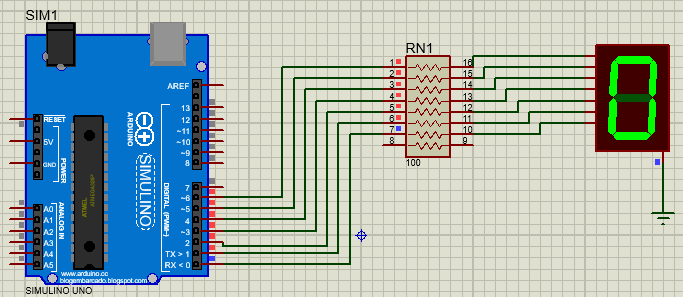
|  |
| --- |
| #include <Wire.h>  #include <LiquidCrystal\_I2C.h>  LiquidCrystal\_I2C lcd(0x27, 16, 2);  #define sensor 10  #define led 7  void setup()  {  Serial.begin(9600);  pinMode(sensor, INPUT);  pinMode(led, OUTPUT);  lcd.init();  lcd.backlight();  lcd.setCursor(0, 0);  lcd.print(" Sound Sensor");  }  void loop()  {  Serial.println(digitalRead(sensor));  digitalWrite(led, digitalRead(sensor));  lcd.setCursor(0, 1);  if (digitalRead(sensor) == 1)  {  lcd.print("Noise");  }  else  {  lcd.print("Quiet");  }  } |

# Bài 10 Led 7 đoạn

## Mô tả

Bài này thực hiện lập trình, mô phỏng mạch Arduino điều khiển đèn led 7 đoạn sáng theo các số từ 0 đến 9, từ các chân 0 đến chân 6 của mạch.

## Sơ đồ



Hình . Mạch Arduino - điều khiển đèn led 7 đoạn

## Linh kiện

* Một mạch Arduino
* Một bo mạch điện trở
* Một đèn led 7 đoạn

## Code chương trình

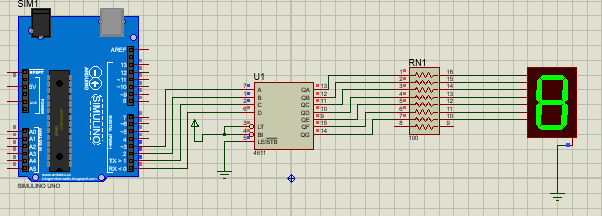
|  |
| --- |
| int a = 6, b=5, c=4, d=3, e=2,f=1, g=0;  void setup(){  pinMode(a,OUTPUT); pinMode(b,OUTPUT); pinMode(c,OUTPUT);  pinMode(d,OUTPUT); pinMode(e,OUTPUT); pinMode(f,OUTPUT); pinMode(g,OUTPUT);  }  void Khong(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, HIGH); digitalWrite(f, HIGH);  digitalWrite(g, LOW);  }  void Mot(){  digitalWrite(a, LOW); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, LOW);digitalWrite(e, LOW);digitalWrite(f, LOW);  digitalWrite(g, LOW);  }  void Hai(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, LOW);  digitalWrite(d, HIGH);digitalWrite(e, HIGH);digitalWrite(f, LOW);  digitalWrite(g, HIGH);  }  void Ba(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, LOW); digitalWrite(f, LOW);  digitalWrite(g, HIGH);  }  void Bon(){  digitalWrite(a, LOW); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, LOW); digitalWrite(e, LOW); digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  }  void Nam(){  digitalWrite(a, HIGH); digitalWrite(b, LOW); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, LOW); digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  }  void Sau(){  digitalWrite(a, HIGH); digitalWrite(b, LOW); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, HIGH); digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  }  void Bay(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, LOW); digitalWrite(e, LOW); digitalWrite(f, LOW);  digitalWrite(g, LOW);  }  void Tam(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, HIGH); digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  }  void Chin(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, LOW); digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  }  void loop(){  Khong();  delay(1000);  Mot();  delay(1000);  Hai();  delay(1000);  Ba();  delay(1000);  Bon();  delay(1000);  Nam();  delay(1000);  Sau();  delay(1000);  Bay();  delay(1000);  Tam();  delay(1000);  Chin();  delay(1000);  } |

# Bài 11 Led 7 đoạn và IC4511

## Mô tả:

Tương tự như bài led 7 đoạn nhưng bài này sử dụng IC 4511 để mở rộng chân

## Sơ đô:



Hình 11.Mạch Arduino-Điều khiển led 7 đoạn ( dùng IC 4511)

## Linh kiện

* Một mạch Arduino
* Một IC4511
* Một bo mạch điện trở
* Một đèn led 7 đoạn chung đất

## Code chương trình

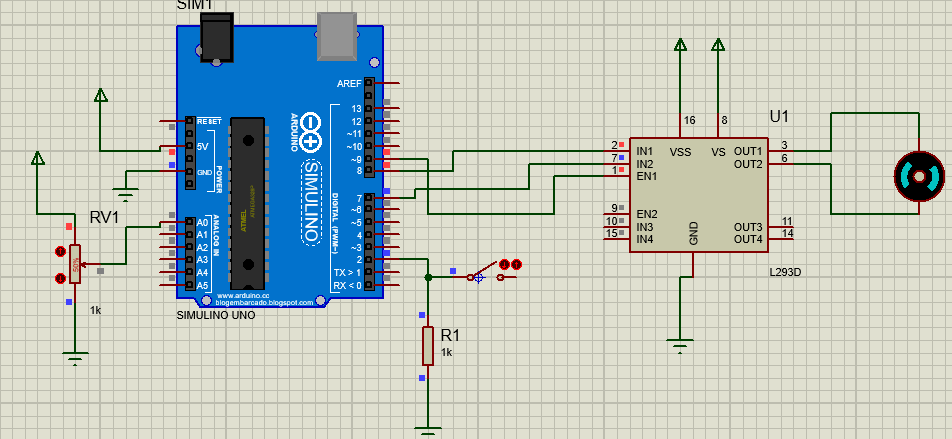
|  |
| --- |
| int A=0, B=1, C=2, D=3;  void setup()  {  pinMode(A,OUTPUT); pinMode(B,OUTPUT);  pinMode(C,OUTPUT); pinMode(D,OUTPUT);  }  void Khong(){  digitalWrite(A,LOW); digitalWrite(B,LOW);  digitalWrite(C,LOW); digitalWrite(D,LOW);  }  void Mot(){  digitalWrite(A,LOW); digitalWrite(B,LOW);  digitalWrite(C,LOW); digitalWrite(D,HIGH);  }  void Hai(){  digitalWrite(A,LOW); digitalWrite(B,LOW);  digitalWrite(C,HIGH); digitalWrite(D,LOW);  }  void Ba(){  digitalWrite(A,LOW); digitalWrite(B,LOW);  digitalWrite(C,HIGH); digitalWrite(D,HIGH);  }  void Bon(){  digitalWrite(A,LOW); digitalWrite(B,HIGH);  digitalWrite(C,LOW); digitalWrite(D,LOW);  }  void Nam(){  digitalWrite(A,LOW); digitalWrite(B,HIGH);  digitalWrite(C,LOW); digitalWrite(D,HIGH);  }  void Sau(){  digitalWrite(A,LOW); digitalWrite(B,HIGH);  digitalWrite(C,HIGH); digitalWrite(D,LOW);  }  void Bay(){  digitalWrite(A,LOW); digitalWrite(B,HIGH);  digitalWrite(C,HIGH); digitalWrite(D,HIGH);  }  void Tam(){  digitalWrite(A,HIGH); digitalWrite(B,LOW);  digitalWrite(C,LOW); digitalWrite(D,LOW);  }  void Chin(){  digitalWrite(A,HIGH); digitalWrite(B,LOW);  digitalWrite(C,LOW); digitalWrite(D,HIGH);  }  void loop(){  Khong(); delay(1000);  Mot(); delay(1000);  Hai(); delay(1000);  Ba(); delay(1000);  Bon(); delay(1000);  Nam(); delay(1000);  Sau(); delay(1000);  Bay(); delay(1000);  Tam(); delay(1000);  Chin(); delay(1000);  } |

# Bài 12: Điều khiển motor

## Mô tả

Bài này thực hiện lập trình, mô phỏng mạch Arduino điều khiển motor bằng L293D

## Sơ đồ



Hình . Mạch Arduino - điểu khiển motor bằng L2932D

## Linh kiện

* Một mạch Arduino
* Một biến trở (POT-HG) 1kΩ
* Một điện trở R1 1kΩ
* Một L293D
* Một motor

## Code chương trình

|  |
| --- |
| #define switchPin 2 // chân công tắc  #define motorPin1 8 // L293D Input 1  #define motorPin2 7 // L293D Input 2  #define speedPin 9 // L293D enable chân 1  #define potPin 0 // chân biến trở nối với A0  int Mspeed = 0;  void setup() {  pinMode(switchPin, INPUT);  pinMode(motorPin1, OUTPUT);  pinMode(motorPin2, OUTPUT);  pinMode(speedPin, OUTPUT);  }  void loop() {  Mspeed = analogRead(potPin)/4;  analogWrite (speedPin, Mspeed);  if (digitalRead(switchPin)) {  digitalWrite(motorPin1, LOW);  digitalWrite(motorPin2, HIGH);  }  else {  digitalWrite(motorPin1, HIGH);  digitalWrite(motorPin2, LOW);  }  } |

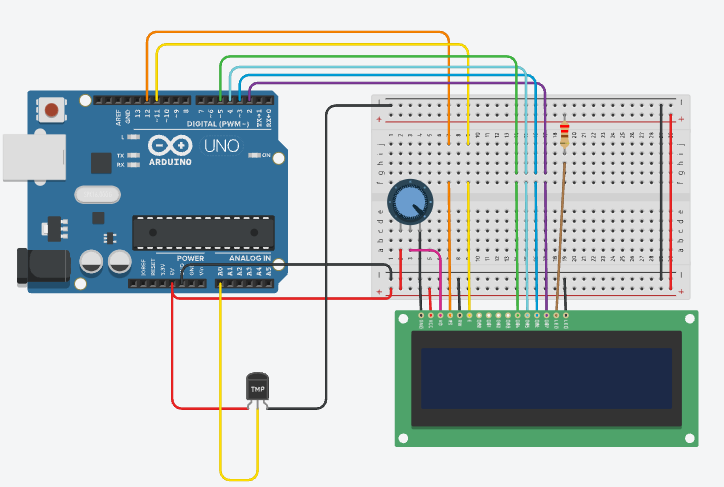
# Bài kiểm tra:

# Bài 1a. Đọc nhiệt độ môi trường và in ra màn hình LCD

## Mô tả:

Bài này sử dụng lập trình, mô phỏng Arduino đọc nhiệt độ bằng TMP qua chân A0, và in kết quả ra màn hình LCD.

## Sơ đồ



Hình 13. Mạch Arduino - đo nhiệt độ TMP và in ra màn hình LCD

## Linh kiên

* Một mạch Arduino
* Một cảm biến nhiệt độ TMP
* Một màn hình LCD
* Một biến trở và một điện trở

## Code chương trình

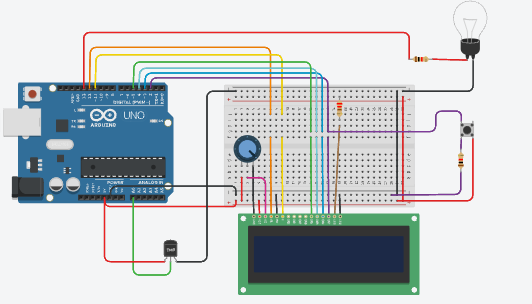
|  |
| --- |
| #include <LiquidCrystal.h>  #include <SoftwareSerial.h>  // initialize the library with the numbers of the interface pins  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  int sensorTMPPin;// chân analog kết nối tới cảm biến LM36  int y = 0;  int pinled = 13;  int button = 2;  void setup() {    pinMode(pinled, OUTPUT);  pinMode(button, INPUT);  Serial.begin(96000);  // set up the LCD's number of columns and rows:  lcd.begin(16, 2);  // Print a message to the LCD.  lcd.print("Nhiet do");  Serial.println("Nhiet do");  }  void loop() {  // set the cursor to column 0, line 1  // (note: line 1 is the second row, since counting begins with 0):  int x = analogRead(A0);  int sensorTMPPin = map ( ((x -20)\*3.04),0,1023, -40, 125);  lcd.setCursor(0, 0);  lcd.print("Nhiet do");  lcd.setCursor(0, 1);  lcd.print(sensorTMPPin);  lcd.print("C"); // print the number of seconds since reset:  Serial.println(sensorTMPPin);  delay( 1000);  lcd.clear();  } |

# Bài 1b: Ngắt đèn nến có ngắt xảy ra

## Mô tả:

Tương tự như câu 1a, và lặp thêm một công tắc điều khiển hoạt động của đèn thông qua chân số 2.

## Sơ đồ



Hình 14.Mạch Arduino\_dieu khien den qua cong tac

## Linh kiện

* Mạch Arduino
* Cảm ứng nhiệt độ TMP,
* Biến trở và hai điện trở
* Màn hình LCD
* Công tắc

## Code chương trình

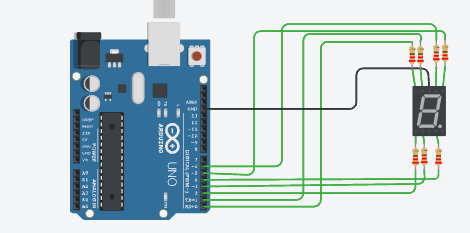
|  |
| --- |
| #include <LiquidCrystal.h>  #include <SoftwareSerial.h>  // initialize the library with the numbers of the interface pins  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  int sensorTMPPin;// chân analog kết nối tới cảm biến LM36  int y = 0;  int pinled = 13;  int button = 2;  void setup() {    pinMode(pinled, OUTPUT);  pinMode(button, INPUT);  Serial.begin(96000);  // set up the LCD's number of columns and rows:  lcd.begin(16, 2);  // Print a message to the LCD.  lcd.print("Nhiet do");  Serial.println("Nhiet do");  }  void loop() {  // set the cursor to column 0, line 1  // (note: line 1 is the second row, since counting begins with 0):  int x = analogRead(A0);  int sensorTMPPin = map ( ((x -20)\*3.04),0,1023, -40, 125);  lcd.setCursor(0, 0);  lcd.print("Nhiet do");  lcd.setCursor(0, 1);  lcd.print(sensorTMPPin);  lcd.print("C"); // print the number of seconds since reset:  Serial.println(sensorTMPPin);  delay( 1000);  lcd.clear();  batden();      }  void batden()  {  y = digitalRead(button);  // kiểm tra xem nút có được nhấn không. Nếu đúng như vậy, x là CAO:  if (y == HIGH) {  // turn LED on:  lcd.clear();  digitalWrite(pinled, HIGH);  } else {  // turn LED off:  digitalWrite(pinled, LOW);  }  delay(10); // độ trễ button  } |

# Bài 2a: Hiện thì số 0 đến 9 bằng led 7 đoạn

## Mô tả

Bài này thực hiện lập trình, mô phỏng mạch Arduino điều khiển đèn led 7 đoạn hiện thị các số từ 0 đến 9 qua các cổng 0 đến cổng 6

## Sơ đồ



Hình 15. Mach Arduino - Điều khiển led 7 đoạn

## Linh kiện

* Mạch Arduino
* 7 điện trở 220Ω
* Một led 7 đoạn chung đất

## Code chương trình

|  |
| --- |
| int a = 6, b=5, c=4, d=3, e=2,f=1, g=0;  void setup(){  pinMode(a,OUTPUT); pinMode(b,OUTPUT); pinMode(c,OUTPUT);  pinMode(d,OUTPUT); pinMode(e,OUTPUT); pinMode(f,OUTPUT); pinMode(g,OUTPUT);  }  void Khong(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, HIGH); digitalWrite(f, HIGH);  digitalWrite(g, LOW);  }  void Mot(){  digitalWrite(a, LOW); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, LOW);digitalWrite(e, LOW);digitalWrite(f, LOW);  digitalWrite(g, LOW);  }  void Hai(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, LOW);  digitalWrite(d, HIGH);digitalWrite(e, HIGH);digitalWrite(f, LOW);  digitalWrite(g, HIGH);  }  void Ba(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, LOW); digitalWrite(f, LOW);  digitalWrite(g, HIGH);  }  void Bon(){  digitalWrite(a, LOW); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, LOW); digitalWrite(e, LOW); digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  }  void Nam(){  digitalWrite(a, HIGH); digitalWrite(b, LOW); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, LOW); digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  }  void Sau(){  digitalWrite(a, HIGH); digitalWrite(b, LOW); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, HIGH); digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  }  void Bay(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, LOW); digitalWrite(e, LOW); digitalWrite(f, LOW);  digitalWrite(g, LOW);  }  void Tam(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, HIGH); digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  }  void Chin(){  digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH);  digitalWrite(d, HIGH); digitalWrite(e, LOW); digitalWrite(f, HIGH);  digitalWrite(g, HIGH);  }  void loop(){  Khong();  delay(1000);  Mot();  delay(1000);  Hai();  delay(1000);  Ba();  delay(1000);  Bon();  delay(1000);  Nam();  delay(1000);  Sau();  delay(1000);  Bay();  delay(1000);  Tam();  delay(1000);  Chin();  delay(1000);  } |

# Bài 2B: Hiện thị từ 0 đến 99 bằng led 7 đoạn

## Mô tả

* Sử dụng hai led 7 đoạn ghép lại với nhau, điều khiển bằng mạch Arduino qua các chân

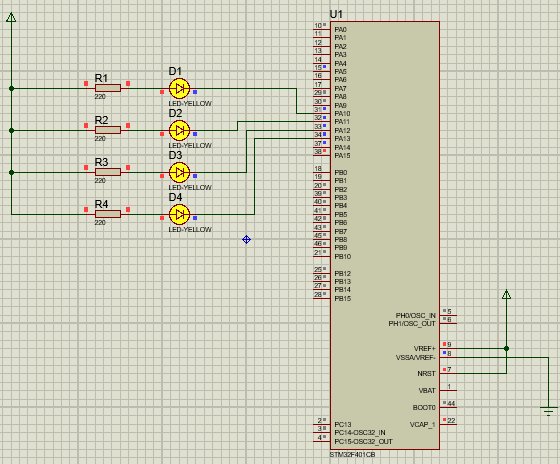
1. STM32F4

# Bài 1: Sử dụng chip STM32F4 401VB nháy 4 led

## Mô tả

Bài này thực hiện lập trình mô phỏng việc điều khiển nhấp nháy 4 đèn led bằng chip STM32F4 401VB

## Sơ đồ



## Linh kiện

* Một chip STM32F4 401 VB
* 4 đèn led
* 4 điện trở có tác dụng cản trở dòng

## Code chương trình

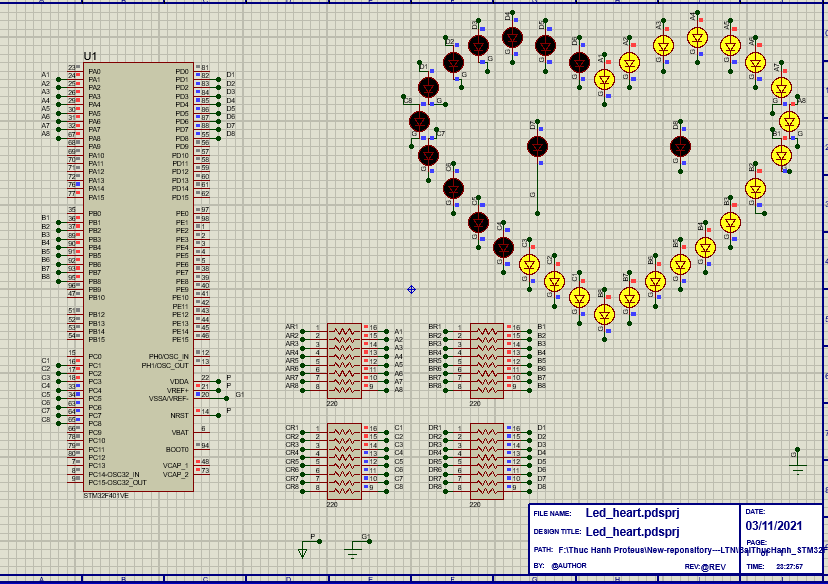
|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  int main(void)  {  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  {  HAL\_GPIO\_TogglePin(GPIOA, GPIO\_PIN\_10| GPIO\_PIN\_11 | GPIO\_PIN\_12| GPIO\_PIN\_13);  HAL\_Delay(1000);    }  }  void SystemClock\_Config(void)  {  RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};  RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};  \_\_HAL\_RCC\_PWR\_CLK\_ENABLE();  \_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE2);  RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI;  RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON;  RCC\_OscInitStruct.HSICalibrationValue = RCC\_HSICALIBRATION\_DEFAULT;  RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;  if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)  {  Error\_Handler();  }  RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK  |RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;  RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_HSI;  RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV2;  RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV1;  RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;  if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_0) != HAL\_OK)  {  Error\_Handler();  }  }  static void MX\_GPIO\_Init(void)  {  GPIO\_InitTypeDef GPIO\_InitStruct = {0};  \_\_HAL\_RCC\_GPIOH\_CLK\_ENABLE();  \_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();  /\*Configure GPIO pin Output Level \*/  HAL\_GPIO\_WritePin(GPIOA, LED\_OUT\_4\_Pin|LED\_OUT\_3\_Pin|LED\_OUT\_2\_Pin|LED\_OUT\_1\_Pin, GPIO\_PIN\_RESET);  /\*Configure GPIO pins : LED\_OUT\_4\_Pin LED\_OUT\_3\_Pin LED\_OUT\_2\_Pin LED\_OUT\_1\_Pin \*/  GPIO\_InitStruct.Pin = LED\_OUT\_4\_Pin|LED\_OUT\_3\_Pin|LED\_OUT\_2\_Pin|LED\_OUT\_1\_Pin;  GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;  GPIO\_InitStruct.Pull = GPIO\_NOPULL;  GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;  HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);  } |

# Bài 2: Sử dụng STM32F4 điều khiển led trái tim.

## Mô tả:

Bài này thực hiện lập trình, mô phỏng STM32F4 điều khiển đèn led trái tim sáng theo chiều kim đồng hồ, chớp nháy, sáng nhứng đèn led chẳn lẻ xen kẽ.

## Sơ đồ:



Hình . STM32F4 điều khiển led trái tim

## Linh kiên:

* Một STM32F4401VE
* Bốn bo mạch điện trở
* Mô hình led trái tim

## Code chương trình

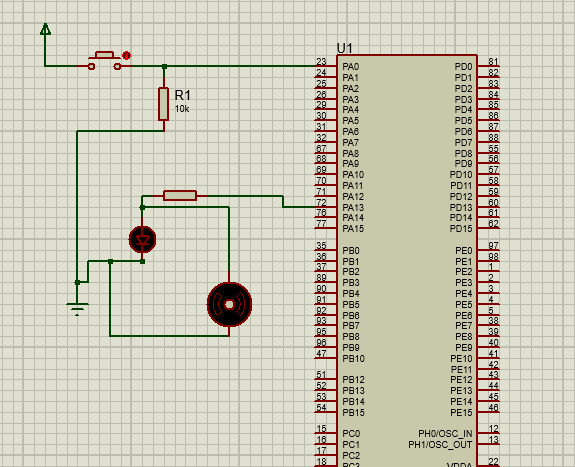
|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  /\*\*  \* @brief The application entry point.  \* @retval int  \*/  void LightAllLed();  void TurnOffAllLed();  void FlashingLed();  void FlashLedOdd();  void FlashEvenLed();  void FlashAllEvenLed();  void FlashAllOddLed();  #define a = [1, 2, 3, 4, 5, 6, 7, 8];  int main(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  {  int x=0;  LightAllLed();  HAL\_Delay(500);  TurnOffAllLed();  HAL\_Delay(500);  FlashingLed();  HAL\_Delay(200);  //-----------------------------------  LightAllLed();  HAL\_Delay(100);  TurnOffAllLed();  HAL\_Delay(100);  //--------------------------------------  FlashLedOdd();  HAL\_Delay(200);  TurnOffAllLed();  HAL\_Delay(200);  FlashEvenLed();  HAL\_Delay(200);  //-------------------------------------  while ( x <10 ){  FlashAllEvenLed();  HAL\_Delay(200);  TurnOffAllLed();  HAL\_Delay(100);  FlashAllOddLed();  HAL\_Delay(200);  TurnOffAllLed();  HAL\_Delay(100);  x++;  }  }  }    void LightAllLed(){  HAL\_GPIO\_WritePin(GPIOA, 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\_SET);  HAL\_GPIO\_WritePin(GPIOB, 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\_SET);  HAL\_GPIO\_WritePin(GPIOC, 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\_SET);  HAL\_GPIO\_WritePin(GPIOD, 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\_SET);  }  void TurnOffAllLed(){  HAL\_GPIO\_WritePin(GPIOA, 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\_RESET);  HAL\_GPIO\_WritePin(GPIOB, 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\_RESET);  HAL\_GPIO\_WritePin(GPIOC, 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\_RESET);  HAL\_GPIO\_WritePin(GPIOD, 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\_RESET);    }  void FlashingLed(){  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_1, GPIO\_PIN\_SET); //1  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_2, GPIO\_PIN\_SET); //2  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_3, GPIO\_PIN\_SET); //3  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, GPIO\_PIN\_SET); //4  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5, GPIO\_PIN\_SET); //5  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, GPIO\_PIN\_SET); //6  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, GPIO\_PIN\_SET); //7  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_8, GPIO\_PIN\_SET); //8  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_1, GPIO\_PIN\_SET); //9  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_2, GPIO\_PIN\_SET); //10  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_3, GPIO\_PIN\_SET); //11  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_4, GPIO\_PIN\_SET); //12  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_5, GPIO\_PIN\_SET); //13  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, GPIO\_PIN\_SET); //14  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_7, GPIO\_PIN\_SET); //15  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_8, GPIO\_PIN\_SET); //16  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_1, GPIO\_PIN\_SET); //17  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_2, GPIO\_PIN\_SET); //18  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_3, GPIO\_PIN\_SET); //19  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_4, GPIO\_PIN\_SET); //20  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_5, GPIO\_PIN\_SET); //21  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_6, GPIO\_PIN\_SET); //22  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_7, GPIO\_PIN\_SET); //23  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_8, GPIO\_PIN\_SET); //24  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET); //25  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET); //26  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET); //27  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET); //28  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET); //29  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET); //30  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET); //31  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET); //32  HAL\_Delay(300);  }  void FlashLedOdd(){  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_1, GPIO\_PIN\_SET); //1  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_3, GPIO\_PIN\_SET); //3  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5, GPIO\_PIN\_SET); //5  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, GPIO\_PIN\_SET); //7  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_1, GPIO\_PIN\_SET); //9  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_3, GPIO\_PIN\_SET); //11  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_5, GPIO\_PIN\_SET); //13  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_7, GPIO\_PIN\_SET); //15  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_1, GPIO\_PIN\_SET); //17  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_3, GPIO\_PIN\_SET); //19  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_5, GPIO\_PIN\_SET); //21  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_7, GPIO\_PIN\_SET); //23  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET); //25  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET); //27  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET); //29  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET); //31  HAL\_Delay(300);  }  void FlashEvenLed(){  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_2, GPIO\_PIN\_SET); //1  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, GPIO\_PIN\_SET); //3  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, GPIO\_PIN\_SET); //5  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_8, GPIO\_PIN\_SET); //7  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_2, GPIO\_PIN\_SET); //1  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_4, GPIO\_PIN\_SET); //3  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, GPIO\_PIN\_SET); //5  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_8, GPIO\_PIN\_SET); //7  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_2, GPIO\_PIN\_SET); //1  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_4, GPIO\_PIN\_SET); //3  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_6, GPIO\_PIN\_SET); //5  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_8, GPIO\_PIN\_SET); //7  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET); //25  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET); //27  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET); //29  HAL\_Delay(300);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET); //31  HAL\_Delay(300);  }  void FlashAllEvenLed(){  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_2, GPIO\_PIN\_SET); //1  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, GPIO\_PIN\_SET); //3  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, GPIO\_PIN\_SET); //5  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_8, GPIO\_PIN\_SET); //7  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_2, GPIO\_PIN\_SET); //1  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_4, GPIO\_PIN\_SET); //3  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, GPIO\_PIN\_SET); //5  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_8, GPIO\_PIN\_SET); //7  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_2, GPIO\_PIN\_SET); //1  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_4, GPIO\_PIN\_SET); //3  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_6, GPIO\_PIN\_SET); //5  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_8, GPIO\_PIN\_SET); //7  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET); //25  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET); //27  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET); //29  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET); //31  } |

# Bài 3: Ngắt ngoài.

## Mô tả:

Bài này thực hiện ngắt ngoài, dọc tín hiệu từ chân A0 để điều khiển việc bật( tắt) đèn hay motor.

## Sơ đồ:



Hình 17. Sử dụng STM32F4VE đọc tín hiệu từ chân A0 để điều khiển tín hiệu chân A13

## Linh kiện

* Một nút bấm
* Hai điện trở dòng
* Một mạch điều khiển STM32F4VE
* Một đèn led
* Một motor

## Code chương trình:

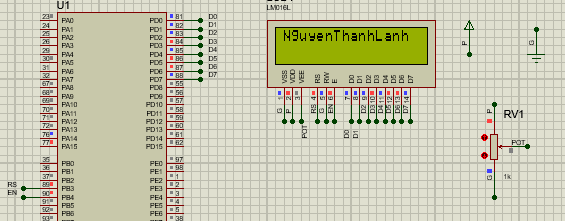
|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  int main(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  {  GPIO\_PinState pinState = HAL\_GPIO\_ReadPin(GPIOA , GPIO\_PIN\_0); //doc chan A0  if( pinState == GPIO\_PIN\_SET)  {  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_13, GPIO\_PIN\_SET); //ban den khi co tin hieu tu chan A0  }  else  {  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_13, GPIO\_PIN\_RESET); //tat den khi co tin hieu tu chan AO  }  }  } |

# Bài 4 :LED LCD

## Mô tả

Tiến hành điều khiển led LCD 16x2 bằng STM32F4 qua cổng D0 đến D7 có sử dụng biến trở điều khiển độ tương phản thông qua cổng VEE

## Sơ đồ



## Linh kiện

* STM32F4VE
* Một POT để điều khiển độ tương phản
* Một LCD 16x2

## Code chương trình

|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  // ham goi 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 khong  //tim gia tri 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 goi lenh  void sendCMD2LCD(char cmd){  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  send8BitLCD(cmd);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_4, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_4, GPIO\_PIN\_SET);  HAL\_Delay(1);  }  // ham goi chuoi ky tu hien thi  void sendcharLCD(char \_char){  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_3, GPIO\_PIN\_SET);  send8BitLCD(\_char);  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++){  sendcharLCD(str[i]);  }  }  int main(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();    // Xóa noi dung LCD  sendCMD2LCD(0x01);  // bat hien thi mang hinh, tat con tro  sendCMD2LCD(0x0C);  // in lên mang hinh LCD chu hello  sendstring2LCD("NguyenThanhLanh");  while (1)  {    }  /\* USER CODE END 3 \*/  }  /\*\*  \* @brief System Clock Configuration  \* @retval None  \*/  void SystemClock\_Config(void)  {  RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};  RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};  /\*\* Configure the main internal regulator output voltage  \*/  \_\_HAL\_RCC\_PWR\_CLK\_ENABLE();  \_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE2);  /\*\* Initializes the RCC Oscillators according to the specified parameters  \* in the RCC\_OscInitTypeDef structure.  \*/  RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI;  RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON;  RCC\_OscInitStruct.HSICalibrationValue = RCC\_HSICALIBRATION\_DEFAULT;  RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;  if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)  {  Error\_Handler();  }  /\*\* Initializes the CPU, AHB and APB buses clocks  \*/  RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK  |RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;  RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_HSI;  RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1;  RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV1;  RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;  if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_0) != HAL\_OK)  {  Error\_Handler();  }  }  static void MX\_GPIO\_Init(void)  {  GPIO\_InitTypeDef GPIO\_InitStruct = {0};  /\* GPIO Ports Clock Enable \*/  \_\_HAL\_RCC\_GPIOD\_CLK\_ENABLE();  \_\_HAL\_RCC\_GPIOB\_CLK\_ENABLE();  /\*Configure GPIO pin Output Level \*/  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\_RESET);  /\*Configure GPIO pin Output Level \*/  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_3|GPIO\_PIN\_4, GPIO\_PIN\_RESET);  /\*Configure GPIO pins : PD0 PD1 PD2 PD3  PD4 PD5 PD6 PD7 \*/  GPIO\_InitStruct.Pin = GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3  |GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7;  GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;  GPIO\_InitStruct.Pull = GPIO\_NOPULL;  GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;  HAL\_GPIO\_Init(GPIOD, &GPIO\_InitStruct);  /\*Configure GPIO pins : PB3 PB4 \*/  GPIO\_InitStruct.Pin = GPIO\_PIN\_3|GPIO\_PIN\_4;  GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;  GPIO\_InitStruct.Pull = GPIO\_NOPULL;  GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;  HAL\_GPIO\_Init(GPIOB, &GPIO\_InitStruct);  }  void Error\_Handler(void)  {  /\* USER CODE BEGIN Error\_Handler\_Debug \*/  /\* User can add his own implementation to report the HAL error return state \*/  /\* USER CODE END Error\_Handler\_Debug \*/  }  #ifdef USE\_FULL\_ASSERT  /\*\*  \* @brief Reports the name of the source file and the source line number  \* where the assert\_param error has occurred.  \* @param file: pointer to the source file name  \* @param line: assert\_param error line source number  \* @retval None  \*/  void assert\_failed(uint8\_t \*file, uint32\_t line)  {  /\* USER CODE BEGIN 6 \*/  /\* User can add his own implementation to report the file name and line number,  tex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) \*/  /\* USER CODE END 6 \*/  }  #endif /\* USE\_FULL\_ASSERT \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* (C) COPYRIGHT STMicroelectronics \*\*\*\*\*END OF FILE\*\*\*\*/ |

**Mục lục**

[I. ARDUINO 1](#_Toc90853554)

[Bài 1. Led nhấp nháy: 1](#_Toc90853555)

[ Mô tả 1](#_Toc90853556)

[ Sơ đồ 1](#_Toc90853557)

[ Linh kiện 1](#_Toc90853558)

[ Code chương trình 1](#_Toc90853559)

[Bài 2. Bật tắt đèn Led bằng nút bấm 2](#_Toc90853560)

[ Mô tả 2](#_Toc90853561)

[ Sơ đồ 2](#_Toc90853562)

[ Linh kiện 2](#_Toc90853563)

[ Code chương trình 2](#_Toc90853564)

[Bài 3: Làm việc với cảm biến nhiệt độ. 3](#_Toc90853565)

[ Mô tả 3](#_Toc90853566)

[ Sơ đồ 3](#_Toc90853567)

[ Linh kiện 3](#_Toc90853568)

[ Code chương trình 3](#_Toc90853569)

[Bài 4: Led sáng dần 4](#_Toc90853570)

[ Mô tả 4](#_Toc90853571)

[ Sơ đồ 4](#_Toc90853572)

[ Linh kiện 4](#_Toc90853573)

[ Code chương trình 5](#_Toc90853574)

[Bài 5: Điều khiển độ sáng thông qua chiết áp 5](#_Toc90853575)

[ Mô tả 5](#_Toc90853576)

[ Sơ đồ 5](#_Toc90853577)

[ Linh kiện 5](#_Toc90853579)

[ Code chương trình 6](#_Toc90853580)

[Bài 6: Điều khiển đèn bằng LDR 6](#_Toc90853581)

[ Mô tả 6](#_Toc90853582)

[ Sơ đồ 6](#_Toc90853583)

[ Linh kiện 7](#_Toc90853584)

[ Code chương trình 7](#_Toc90853585)

[Bài 7: LED sáng dần từ đèn 1 tới đèn 10 và ngược lại 7](#_Toc90853586)

[ Mô tả 7](#_Toc90853587)

[ Sơ đồ 8](#_Toc90853588)

[ Linh kiện 8](#_Toc90853589)

[ Code chương trình 8](#_Toc90853590)

[Bài 8: LED sáng dần từ 1 đến 10 và ngược lại, có sử dụng biến trở. 9](#_Toc90853591)

[ Mô tả 9](#_Toc90853592)

[ Sơ đồ 9](#_Toc90853593)

[ Linh kiện 9](#_Toc90853594)

[ Code chương trình 9](#_Toc90853595)

[Bài 9: Cảm biến âm thanh 10](#_Toc90853596)

[ Mô tả 10](#_Toc90853597)

[ Sơ đồ 11](#_Toc90853598)

[ Linh kiện 11](#_Toc90853599)

[ Code chương trình 11](#_Toc90853600)

[Bài 10 Led 7 đoạn 12](#_Toc90853601)

[ Mô tả 12](#_Toc90853602)

[ Sơ đồ 12](#_Toc90853603)

[ Linh kiện 13](#_Toc90853604)

[ Code chương trình 13](#_Toc90853605)

[Bài 11 Led 7 đoạn và IC4511 15](#_Toc90853606)

[ Mô tả: 15](#_Toc90853607)

[ Sơ đô: 15](#_Toc90853608)

[ Linh kiện 15](#_Toc90853609)

[ Code chương trình 15](#_Toc90853610)

[Bài 12: Điều khiển motor 17](#_Toc90853611)

[ Mô tả 17](#_Toc90853612)

[ Sơ đồ 17](#_Toc90853613)

[ Linh kiện 17](#_Toc90853614)

[ Code chương trình 17](#_Toc90853615)

[Bài kiểm tra: 18](#_Toc90853616)

[Bài 1a. Đọc nhiệt độ môi trường và in ra màn hình LCD 18](#_Toc90853617)

[ Mô tả: 18](#_Toc90853618)

[ Sơ đồ 19](#_Toc90853619)

[ Linh kiên 19](#_Toc90853620)

[ Code chương trình 19](#_Toc90853621)

[Bài 1b: Ngắt đèn nến có ngắt xảy ra 20](#_Toc90853622)

[ Mô tả: 20](#_Toc90853623)

[ Sơ đồ 20](#_Toc90853624)

[ Linh kiện 20](#_Toc90853625)

[ Code chương trình 21](#_Toc90853626)

[Bài 2a: Hiện thì số 0 đến 9 bằng led 7 đoạn 22](#_Toc90853627)

[ Mô tả 22](#_Toc90853628)

[ Sơ đồ 22](#_Toc90853629)

[ Linh kiện 22](#_Toc90853630)

[ Code chương trình 23](#_Toc90853631)

[Bài 2B: Hiện thị từ 0 đến 99 bằng led 7 đoạn 25](#_Toc90853632)

[ Mô tả 25](#_Toc90853633)

[II. STM32F4 25](#_Toc90853634)

[Bài 1: Sử dụng chip STM32F4 401VB nháy 4 led 25](#_Toc90853635)

[ Mô tả 25](#_Toc90853636)

[ Sơ đồ 25](#_Toc90853637)

[ Linh kiện 25](#_Toc90853638)

[ Code chương trình 25](#_Toc90853639)

[Bài 2: Sử dụng STM32F4 điều khiển led trái tim. 27](#_Toc90853640)

[ Mô tả: 27](#_Toc90853641)

[ Sơ đồ: 28](#_Toc90853642)

[ Linh kiên: 28](#_Toc90853643)

[ Code chương trình 28](#_Toc90853644)

[Bài 3: Ngắt ngoài. 33](#_Toc90853645)

[ Mô tả: 33](#_Toc90853646)

[ Sơ đồ: 33](#_Toc90853647)

[ Linh kiện 33](#_Toc90853648)

[ Code chương trình: 33](#_Toc90853649)

[Bài 4 :LED LCD 34](#_Toc90853650)

[ Mô tả 34](#_Toc90853651)

[ Sơ đồ 34](#_Toc90853652)

[ Linh kiện 34](#_Toc90853653)

[ Code chương trình 34](#_Toc90853654)