BỘ GIÁO DỤC VÀ ĐẠO TẠO

ĐẠI HỌC NHA TRANG

KHOA CÔNG NGHỆ THÔNG TIN

**BÁO CÁO**

LẬP TRÌNH THIẾT BỊ NHÚNG



Giáo viên hướng dẫn: **Mai Cường Thọ**

Sinh viên thực hiện: **Nguyễn Trọng Hiếu**

*Ngày 27 tháng 9 năm 2021*

# MỤC LỤC

[MỤC LỤC 2](#_Toc88241257)

[BÀI 1 - NHÁY ĐÈN LED CỔNG 13 3](#_Toc88241258)

[BÀI 2 - NHÁY ĐÈN LED CỔNG 13 KHI BẤM BUTTON 4](#_Toc88241259)

[BÀI 3 – ĐỌC NHIỆT ĐỘ, SÁNG ĐÈN LED KHI QUÁ 37˚C 5](#_Toc88241260)

[BÀI 4 – ĐÈN LED SÁNG DẦN 6](#_Toc88241261)

[BÀI 5 – ĐIỀU KHIỂN ĐỘ SÁNG LED QUA CHIẾT ÁP 7](#_Toc88241262)

[BÀI 6 – QUANG TRỞ 1 ĐÈN LED 8](#_Toc88241263)

[BÀI 6.1 – QUANG TRỞ 4 ĐÈN LED 9](#_Toc88241264)

[BÀI 7 – CẢM BIẾN KHOẢNG CÁCH 10](#_Toc88241265)

[BÀI 8 – LED 7 ĐOẠN 11](#_Toc88241266)

[BÀI 9 – DC MOTOR 13](#_Toc88241267)

[BÀI 10 – ĐIỀU KHIỂN 8 LED SÁNG DẦN VÀ TẮT DẦN 14](#_Toc88241268)

[BÀI 11 – HIỂN THỊ NHIỆT ĐỘ MÔI TRƯỜNG RA LCD 15](#_Toc88241269)

[BÀI 12 – NHÁY 4 LED STM32 16](#_Toc88241270)

[BÀI 13 – LED TRÁI TIM STM32 17](#_Toc88241271)

[BÀI 14 – DC VỚI BUTTON STM32 18](#_Toc88241272)

[BÀI 15 – LED VỚI BUTTON STM32 19](#_Toc88241273)

[BÀI 16 – LẬP TRÌNH NGẮT STM32 20](#_Toc88241274)

[BÀI 17 – CẢM BIẾN NHIỆT ĐỘ STM32 21](#_Toc88241275)

[BÀI 18 – LCD STM32 22](#_Toc88241276)

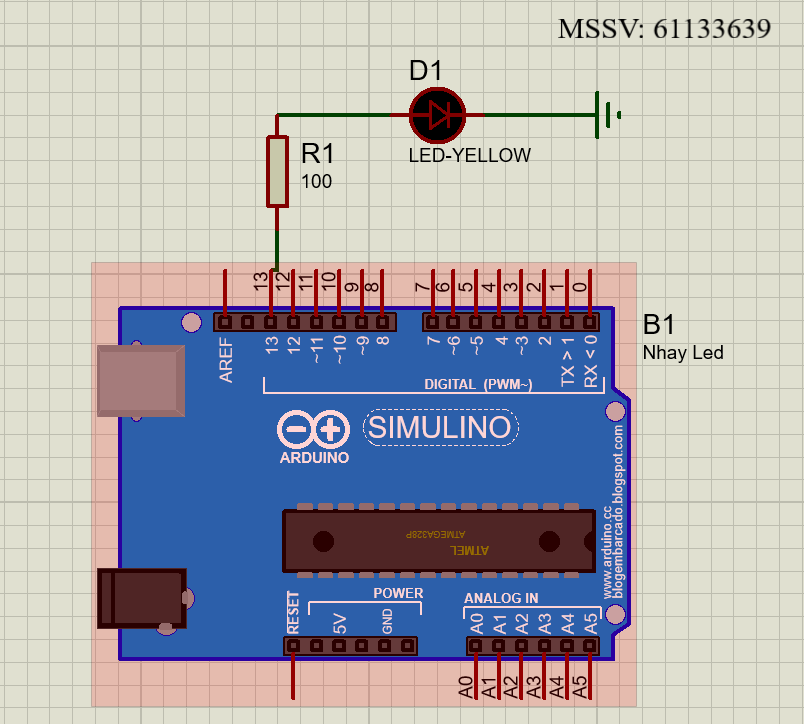
# 

# BÀI 1 - NHÁY ĐÈN LED CỔNG 13

## Mô tả:

Ở bài này, yêu cầu đèn led nháy tự động sau khoản thời gian 1 giây. Đèn led được đấu vào cổng số 13 của Board.

## Sơ đồ thiết kế:

****

Hình 1: Nháy Led

## Đặc điểm linh kiện:

* 1 Mạch Arduino UNO.
* 1 Đèn Led.
* 1 Điện trở: R1 (100Ω).

## Mã lệnh chính:

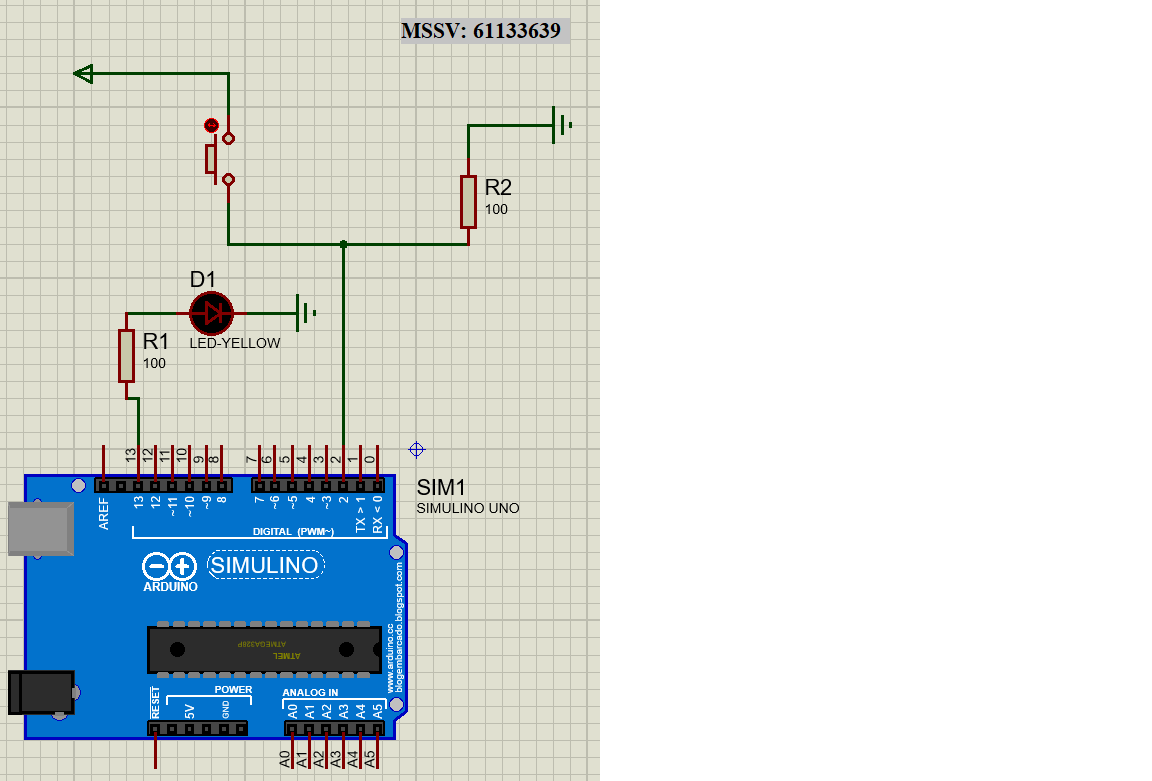
|  |
| --- |
| void **setup**()  {  **pinMode**(13, **OUTPUT**);  }  void **loop**()  {  digitalWrite(13, **HIGH**);  **delay**(1000);  digitalWrite(13, **LOW**);  **delay**(1000);  } |

# BÀI 2 - NHÁY ĐÈN LED CỔNG 13 KHI BẤM BUTTON

## Mô tả:

Ở bài này, yêu cầu đèn led bật/tắt sau khi bấm công tắc. Đèn led được đấu vào cổng số 13 của Board mạch.

## Sơ đồ thiết kế:

****

Hình 2: Nháy Led + Button

## Đặc điểm linh kiện:

* 1 Mạch Arduino UNO.
* 1 Đèn led
* 2 Điện trở: R1 (100Ω), R2 (100Ω).
* 1 Button.

## Mã lệnh chính:

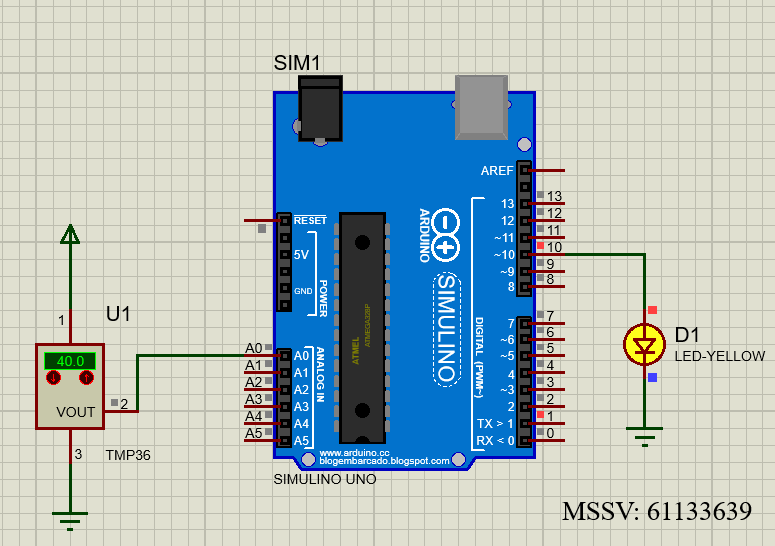
|  |
| --- |
| int **x** = 0;  void **setup**() {  pinMode(2, **INPUT**);  pinMode(13, **OUTPUT**);  }  void **loop**() {  **x** = digitalRead(2);  **if** (**x** == **HIGH**){  digitalWrite(13, **HIGH**);  }  **else**{  digitalWrite(13, **LOW**);  }  **delay**(1000);  } |

# BÀI 3 – ĐỌC NHIỆT ĐỘ, SÁNG ĐÈN LED KHI QUÁ 37˚C

## Mô tả:

Ở bài này, yêu cầu đọc nhiệt độ bằng cảm biến nhiệt độ TMP36 và sáng đèn led nếu nhiệt độ vượt quá 37**˚**C. Đèn led được đấu vào cổng số 10 và cảm biến gắn vào cổng A0 của Board mạch.

## Sơ đồ thiết kế:

****

Hình 3: Cảm biến nhiệt độ

## Đặc điểm linh kiện:

* 1 Mạch Arduino UNO.
* 1 Đèn Led.
* 1 Cảm biến nhiệt độ TMP36.

## Mã lệnh chính:

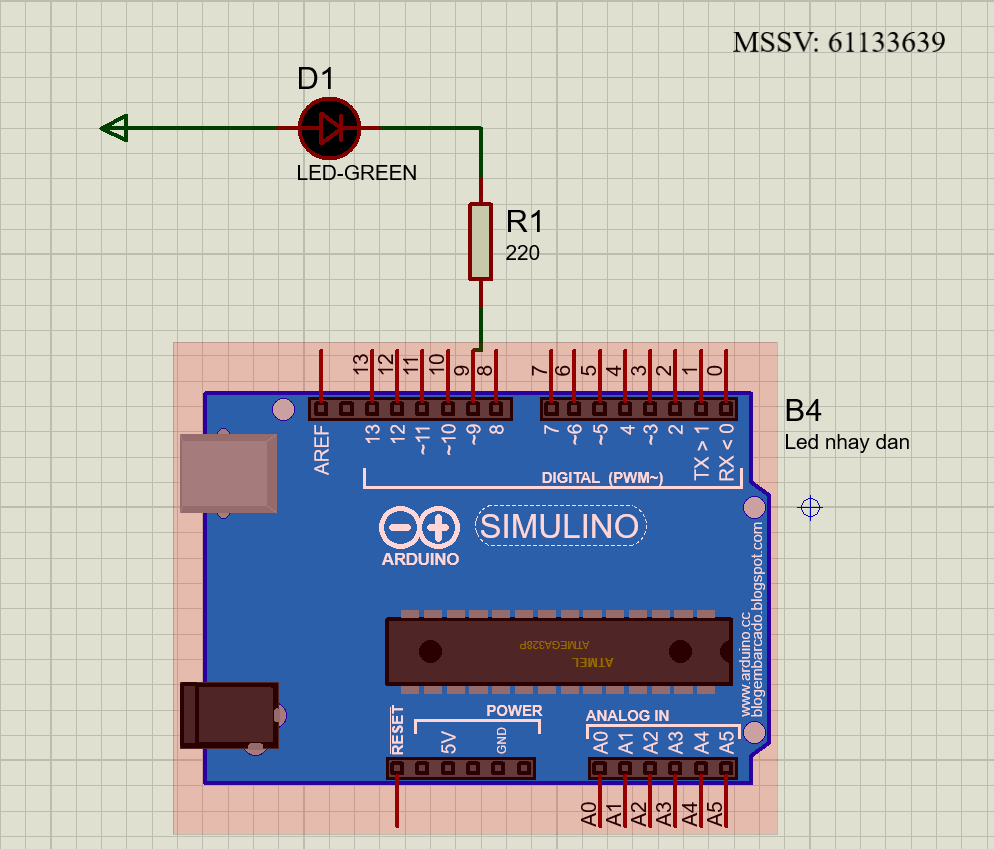
|  |
| --- |
| void **setup**()  {  Serial.**begin**(9600);  }  void **loop**()  {  int **giaTri** = analogRead(A0);  int nhietDo = **map**(**giaTri**, 20, 358, -40, 125);  if(nhietDo > 37) digitalWrite(10, **HIGH**);  else  digital**Write**(10, **LOW**)  **delay**(1000);  } |

# BÀI 4 – ĐÈN LED SÁNG DẦN

## Mô tả:

Ở bài này, yêu cầu 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 thiết kế:

****

Hình 4: Led sáng dần

## Đặc điểm linh kiện:

* 1 Mạch Arduino UNO.
* 1 Đèn Led.
* 1 Điện trở: R1 (220Ω).

## Mã lệnh chính:

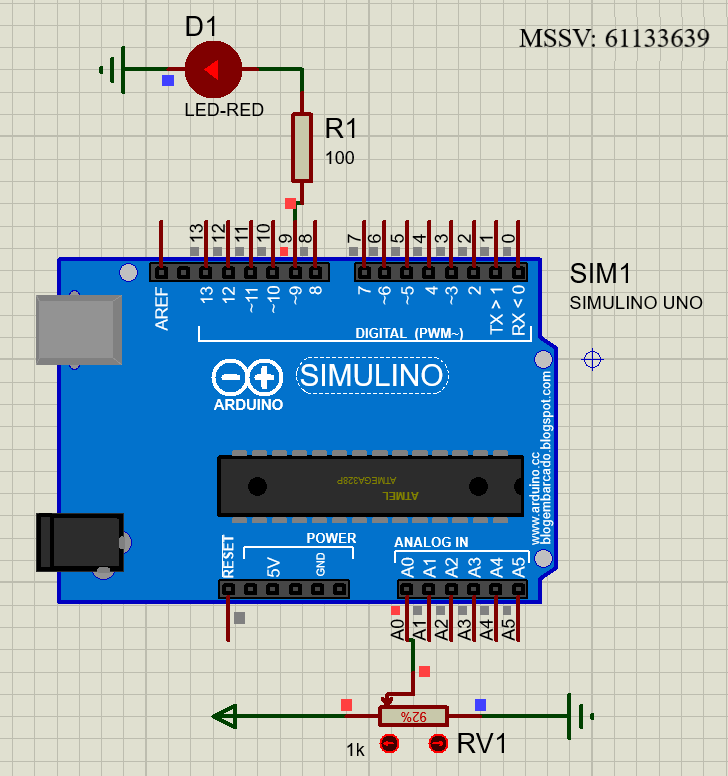
|  |
| --- |
| **int** brs = 0;  void **setup**()  {  pin**Mode**(9, OUTPUT);  }  void **loop**()  {  for (brs = 0; brs <= 255; brs += 5){  analogWrite(9, brs);  **delay**(30);  }  for (brs = 225; brs >= 0; brs -=5){  analogWrite(9, brs);  **delay**(30);  }  } |

# BÀI 5 – ĐIỀU KHIỂN ĐỘ SÁNG LED QUA CHIẾT ÁP

## Mô tả:

Ở bài này, yêu cầu thiết kế hệ thống nhúng cho phép điều khiển độ sáng của Led (đấu cổng 9) thông qua một biến trở (đấu ở chân A0).

## Sơ đồ thiết kế:

****

Hình 5: Điều khiển Led qua chiết áp

## Đặc điểm linh kiện:

* 1 Mạch Arduino UNO.
* 1 Đèn Led.
* 1 Biến trở.
* 1 Điện trở (100Ω).

## Mã lệnh chính:

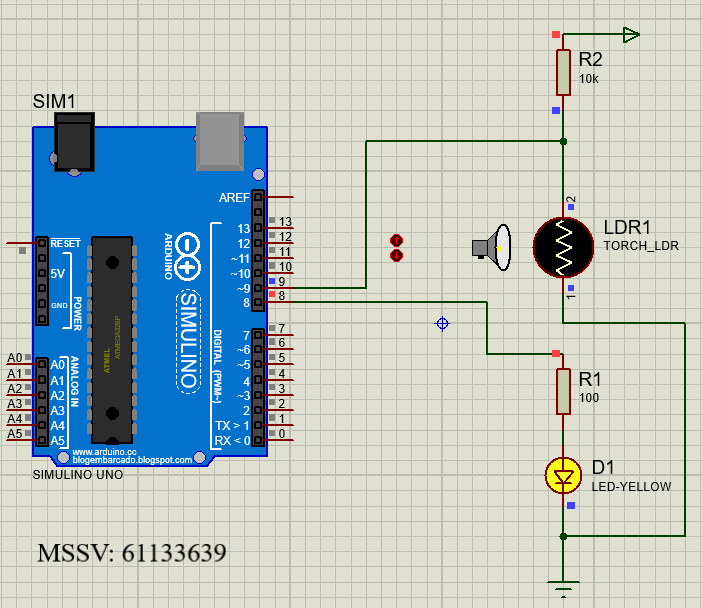
|  |
| --- |
| **int** x = 0;  void **setup**()  {  pin**Mode**(A0, INPUT);  pin**Mode**(9, OUTPUT);  }  void **loop**()  {  X = analog**Read**(A0);  Int brightness = **map**(x,0,1023,0,225);  analog**Write**(9, brightness);  } |

# BÀI 6 – QUANG TRỞ 1 ĐÈN LED

## Mô tả:

Ở bài này, yêu cầu lắp mạch gồm quang trở và Led, khi có ánh sáng chiếu vào quang trở, đèn Led sẽ sáng.

## Sơ đồ thiết kế:

****

Hình 6: Quang trở và đèn led.

## Đặc điểm linh kiện:

* 1 Mạch Arduino UNO.
* 1 Quang trở.
* 2 Điện trở: R1 (100 Ω) và R2 (10k Ω).

## Mã lệnh chính:

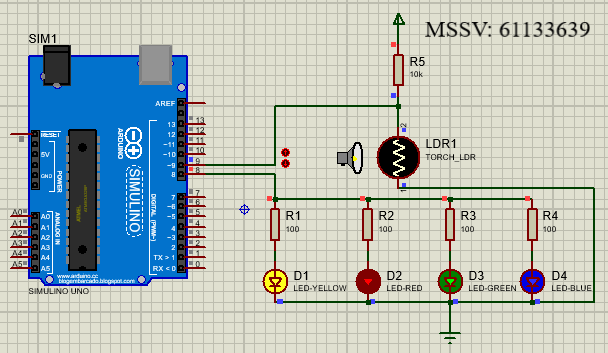
|  |
| --- |
| int **ldr** = 9;  int **led** = 8;  void **setup**()  {  pin**Mode**(ldr, INPUT\_PULLUP);  pin**Mode**(led, OUTPUT);  }  void **loop**()  {  if(digital**Read**(ldr) == HIGH)  {  digital**Write**(led, LOW);  }  else  {  digital**Write**(led, HIGH);  }  } |

# BÀI 6.1 – QUANG TRỞ 4 ĐÈN LED

## Mô tả:

Ở bài này, yêu cầu lắp mạch gồm quang trở và 4 đèn Led, khi có ánh sáng chiếu vào quang trở, 4 đèn Led sẽ sáng.

## Sơ đồ thiết kế:

****

Hình 6: Quang trở và đèn led.

## Đặc điểm linh kiện:

* 1 Mạch Arduino UNO.
* 1 Quang trở.
* 4 Đèn led
* 5 Điện trở: R1, R2, R3, R4 (100 Ω) và R5 (10k Ω).

## Mã lệnh chính:

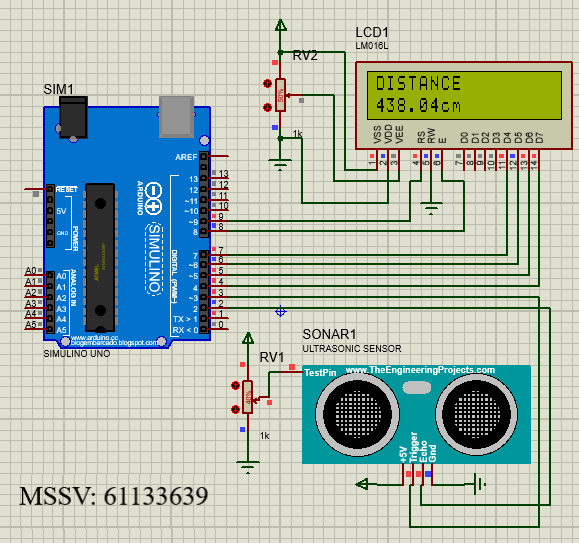
|  |
| --- |
| int **ldr** = 9;  int **led** = 8;  void **setup**()  {  pin**Mode**(ldr, INPUT\_PULLUP);  pin**Mode**(led, OUTPUT);  }  void **loop**()  {  if(digital**Read**(ldr) == HIGH)  {  digital**Write**(led, LOW);  }  else  {  digital**Write**(led, HIGH);  }  } |

# BÀI 7 – CẢM BIẾN KHOẢNG CÁCH

## Mô tả:

Ở bài này, yêu cầu lắp cảm biến Ultrasonic Sensor vào Board mạch rồi hiển thị khoảng cách lên màn hình LCD, khoảng cách có thể điều chỉnh được thông qua biến trở.

## Sơ đồ thiết kế:

****

Hình 7: Cảm biến khoảng cách

## Đặc điểm linh kiện:

* 1 Mạch Arduino UNO.
* 1 Cảm biến Ultrasonic Sensor.
* 1 Biến trở.
* 1 Màn hình LCD.

## Mã lệnh chính:

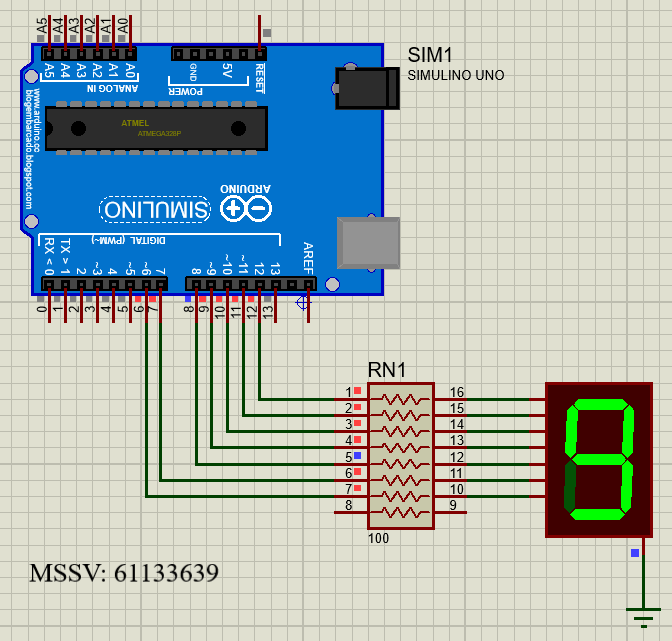
|  |
| --- |
| #include <**LiquidCrystal**.h>  **LiquidCrystal** lcd(9, 8, 7, 6, 5, 4);  const int **trig** = 3;  const int **echo** = 2;  void **setup**() {  Serial.**begin**(9600);  pinMode(**trig**, OUTPUT);  pin**Mode**(**echo**, INPUT);  lcd.**begin**(16,2);  }  void **loop**() {  float **duration**; //Thời gian  float **distance**; //Khoảng cách  digital**Write**(trig, LOW);  delay**Microseconds**(2);  digital**Write**(**trig**, HIGH);  delay**Microseconds**(10);  digital**Write**(trig, HIGH);  duration = pulseIn(**echo**, HIGH, 30000);  distance = (**duration**\*0.034/2);  lcd.set**Cursor**(0,0);  lcd.print("DISTANCE");  lcd.set**Cursor**(0,1);  lcd.print(distance);  lcd.print("cm");  delay(1000);  lcd.**clear**();  } |

# BÀI 8 – LED 7 ĐOẠN

## Mô tả:

Ở bài này, yêu cầu lắp đèn Led 7 đoạn vào Board mạch Arduino thông qua điện trở và hiện lần lượt từ số 0 đến số 9.

## Sơ đồ thiết kế:

****

Hình 8: Led 7 đoạn

## Đặc điểm linh kiện:

* 1 Mạch Arduino UNO.
* 1 Đèn Led 7 đoạn.
* 1 Điện trở (100 Ω).

## Mã lệnh chính:

|  |
| --- |
| int **A** = 12; int **B** = 11; int **C** = 10; int **D** = 9;  int **E** = 8; int **F** = 7; int **G** = 6;  void setup()  {  pinMode(**A**, OUTPUT); pinMode(**B**, OUTPUT); pinMode(**C**, OUTPUT);  pinMode(**D**, OUTPUT); pinMode(**E**, OUTPUT); pinMode(**F**, OUTPUT);  pinMode(**G**, OUTPUT);  }  void loop()  {  //So 0  digitalWrite(A, HIGH); digitalWrite(B, HIGH);  digitalWrite(C, HIGH); digitalWrite(D, HIGH);  digitalWrite(E, HIGH); digitalWrite(F, HIGH);  digitalWrite(G, LOW);  delay(1000);  //So 1  digitalWrite(A, LOW); digitalWrite(B, HIGH);  digitalWrite(C, HIGH); digitalWrite(D, LOW);  digitalWrite(E, LOW); digitalWrite(F, LOW);  digitalWrite(G, LOW);  delay(1000);  //So 2  digitalWrite(A, HIGH); digitalWrite(B, HIGH);  digitalWrite(C, LOW); digitalWrite(D, HIGH);  digitalWrite(E, HIGH); digitalWrite(F, LOW);  digitalWrite(G, HIGH);  delay(1000);  //So 3  digitalWrite(A, HIGH); digitalWrite(B, HIGH);  digitalWrite(C, HIGH); digitalWrite(D, HIGH);  digitalWrite(E, LOW); digitalWrite(F, LOW);  digitalWrite(G, HIGH);  delay(1000);  //So 4  digitalWrite(A, LOW); digitalWrite(B, HIGH);  digitalWrite(C, HIGH); digitalWrite(D, LOW);  digitalWrite(E, LOW); digitalWrite(F, HIGH);  digitalWrite(G, HIGH);  delay(1000);  //So 5  digitalWrite(A, HIGH); digitalWrite(B, LOW);  digitalWrite(C, HIGH); digitalWrite(D, HIGH);  digitalWrite(E, LOW); digitalWrite(F, HIGH);  digitalWrite(G, HIGH);  delay(1000);  //So 6  digitalWrite(A, HIGH); digitalWrite(B, LOW);  digitalWrite(C, HIGH); digitalWrite(D, HIGH);  digitalWrite(E, HIGH); digitalWrite(F, HIGH);  digitalWrite(G, HIGH);  delay(1000);  //So 7  digitalWrite(A, HIGH); digitalWrite(B, HIGH);  digitalWrite(C, HIGH); digitalWrite(D, LOW);  digitalWrite(E, LOW); digitalWrite(F, LOW);  digitalWrite(G, LOW);  delay(1000);  //So 8  digitalWrite(A, HIGH); digitalWrite(B, HIGH);  digitalWrite(C, HIGH); digitalWrite(D, HIGH);  digitalWrite(E, HIGH); digitalWrite(F, HIGH);  digitalWrite(G, HIGH);  delay(1000);  //So 9  digitalWrite(A, HIGH); digitalWrite(B, HIGH);  digitalWrite(C, HIGH); digitalWrite(D, HIGH);  digitalWrite(E, LOW); digitalWrite(F, HIGH);  digitalWrite(G, HIGH);  delay(1000);  } |

# BÀI 9 – DC MOTOR

## Mô tả:

Ở bài này, yêu cầu

## Sơ đồ thiết kế:

## Đặc điểm linh kiện:

* DC Motor.
* Biến trở.

## Mã lệnh chính:

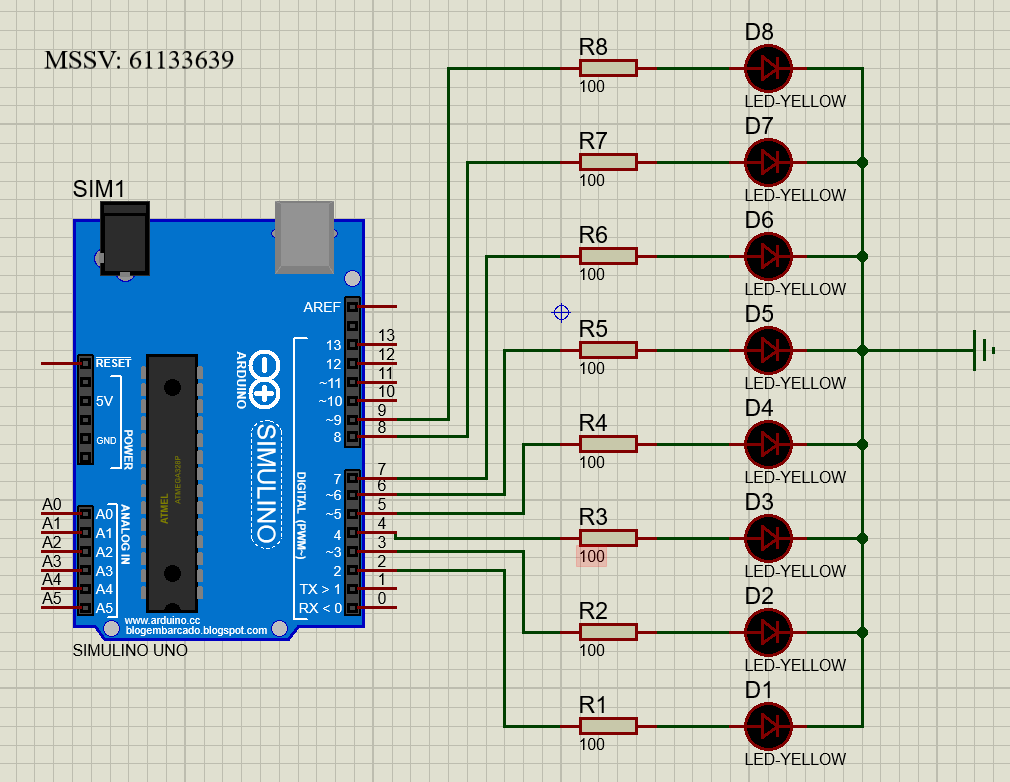
|  |
| --- |
| void **setup**()  {    }  void **loop**()  {    } |

# BÀI 10 – ĐIỀU KHIỂN 8 LED SÁNG DẦN VÀ TẮT DẦN

## Mô tả:

Ở bài này, yêu cầu lắp 8 đèn Led cào Board mạch Arduino và điều khiển Led sáng dần và tắt dần sau mỗi giây.

## Sơ đồ thiết kế:

****

## Đặc điểm linh kiện:

* 1 Mạch Arduino UNO.
* 8 Đèn Led.
* 8 Điện trở mỗi điện trở 100 Ω.

## Mã lệnh chính:

|  |
| --- |
| byte **ledPin**[] = {2,3,4,5,6,7,8,9};  byte **pinCount**;  void **setup**() {  pin**Count** = sizeof(led**Pin**);  for (int i=0; i<pin**Count**; i++) {  pin**Mode**(led**Pin**[i],OUTPUT);  digital**Write**(ledPin[i],LOW);  }  }  void **loop**() {  for (int i=0; i < pin**Count**; i++) {  digital**Write**(ledPin[i],HIGH);  **delay**(1000);  }  for (int i=0; i<pin**Count**; i+=1) {  digital**Write**(ledPin[i],LOW);  **delay**(1000);  }  } |

# BÀI 11 – HIỂN THỊ NHIỆT ĐỘ MÔI TRƯỜNG RA LCD

## Mô tả:

Ở bài này, yêu cầu dùng cảm biến nhiệt độ TMP36 để đọc nhiệt độ môi trường rồi hiển thị nhiệt độ đọc được ra màn hình LCD sau mỗi giây.

## Sơ đồ thiết kế:

## Đặc điểm linh kiện:

* DC Motor.
* Biến trở.

## Mã lệnh chính:

|  |
| --- |
| **int** speedControl = 9;  int inp1 = 8;  int inp2 = 7;  void **setup**()  {  pinMode(speedControl, OUTPUT);  pinMode(inp1, OUTPUT);  pinMode(inp2, OUTPUT);  }  void **loop**()  {  digitalWrite(inp1, HIGH);  digitalWrite(inp2, HIGH);  analogWrite(speedControl, 127);  } |

# BÀI 12 – NHÁY 4 LED STM32

## Mô tả:

Ở bài này, yêu cầu dùng cảm biến nhiệt độ TMP36 để đọc nhiệt độ môi trường rồi hiển thị nhiệt độ đọc được ra màn hình LCD sau mỗi giây.

## Sơ đồ thiết kế:

## Đặc điểm linh kiện:

* DC Motor.
* Biến trở.

## Mã lệnh chính:

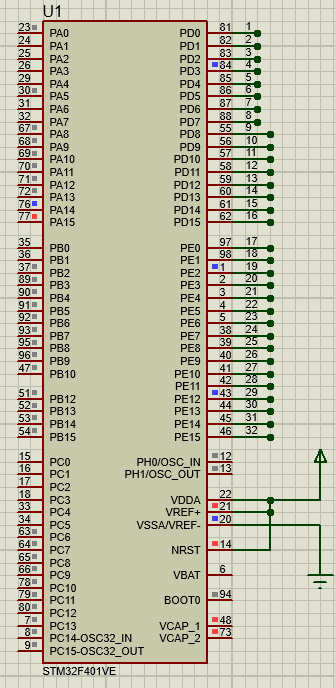
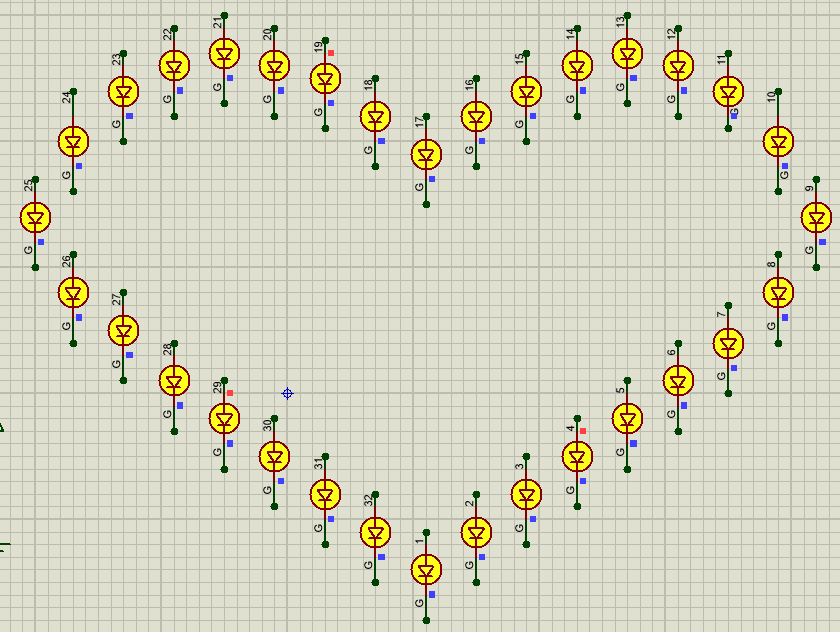
|  |
| --- |
| **int** speedControl = 9;  int inp1 = 8;  int inp2 = 7;  void **setup**()  {  pinMode(speedControl, OUTPUT);  pinMode(inp1, OUTPUT);  pinMode(inp2, OUTPUT);  }  void **loop**()  {  digitalWrite(inp1, HIGH);  digitalWrite(inp2, HIGH);  analogWrite(speedControl, 127);  } |

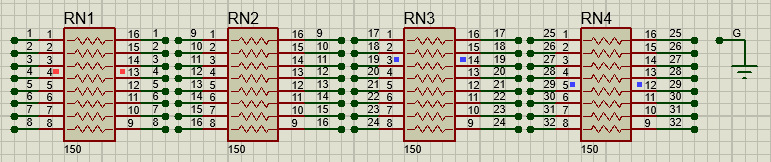
# BÀI 13 – LED TRÁI TIM STM32

## Mô tả:

Ở bài này, yêu cầu dùng cảm biến nhiệt độ TMP36 để đọc nhiệt độ môi trường rồi hiển thị nhiệt độ đọc được ra màn hình LCD sau mỗi giây.

## Sơ đồ thiết kế:





## Đặc điểm linh kiện:

* DC Motor.
* Biến trở.

## Mã lệnh chính:

|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  void SangHetLed();  void TatHetLed();  void SangLL();  int main(void){  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1){  SangHetLed();  HAL\_Delay(1000);  TatHetLed();  HAL\_Delay(1000);  SangLL();  TatHetLed();  HAL\_Delay(1000);  }  }  void SangHetLed(){  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);    HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_4, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_5, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_6, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_7, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_8, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_9, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_10, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_11, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_12, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_13, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_14, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_15, GPIO\_PIN\_SET);  }  void TatHetLed(){  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_RESET);  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\_12, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_RESET);    HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_4, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_5, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_6, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_7, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_8, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_9, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_10, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_11, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_12, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_13, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_14, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_15, GPIO\_PIN\_RESET);  }  void SangLL(){  TatHetLed();  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);  HAL\_Delay(100);    HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_4, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_5, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_6, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_7, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_8, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_9, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_10, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_11, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_12, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_13, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_14, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_15, GPIO\_PIN\_SET);  HAL\_Delay(100);  } |

# BÀI 14 – DC VỚI BUTTON STM32

## Mô tả:

Ở bài này, yêu cầu

## Sơ đồ thiết kế:

## Đặc điểm linh kiện:

* DC Motor.
* Biến trở.

## Mã lệnh chính:

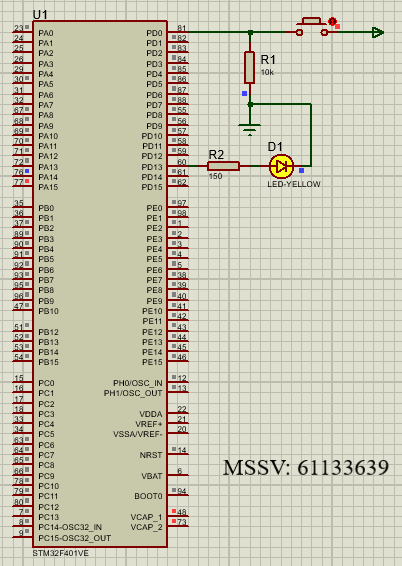
|  |
| --- |
| void **setup**()  {    }  void **loop**()  {    } |

# BÀI 15 – LED VỚI BUTTON STM32

## Mô tả:

Ở bài này, yêu cầu

## Sơ đồ thiết kế:

****

## Đặc điểm linh kiện:

* DC Motor.
* Biến trở.

## Mã lệnh chính:

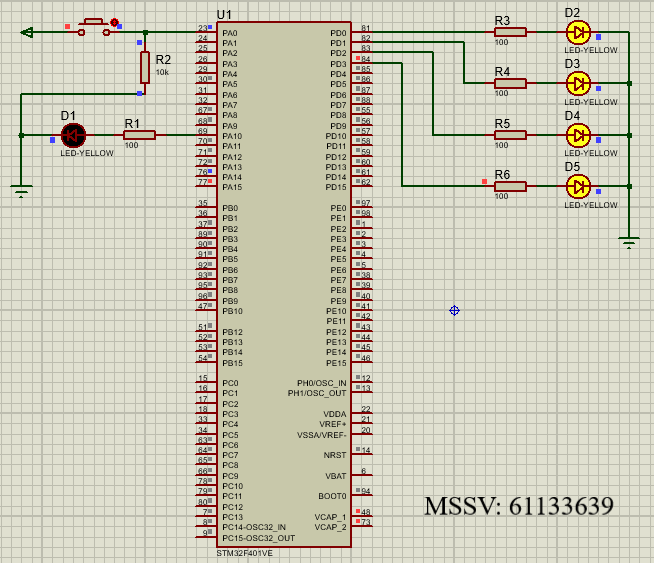
|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  int main(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  {  GPIO\_PinState pin0State = HAL\_GPIO\_ReadPin(GPIOD, GPIO\_PIN\_0);  if (pin0State == GPIO\_PIN\_SET)  {  HAL\_GPIO\_WritePin(GPIOD,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

# BÀI 16 – LẬP TRÌNH NGẮT STM32

## Mô tả:

Ở bài này, yêu cầu

## Sơ đồ thiết kế:

****

## Đặc điểm linh kiện:

* DC Motor.
* Biến trở.

## Mã lệnh chính:

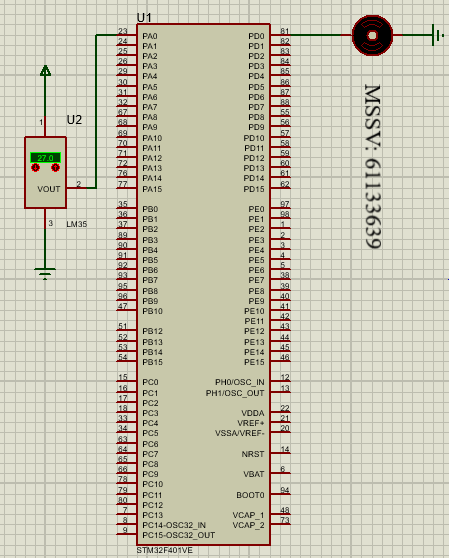
|  |
| --- |
| **#include** "main.h"  **void** SystemClock\_Config(**void**);  **static void** MX\_GPIO\_Init(**void**);  int main(**void**){  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  **while** (1){  **HAL\_GPIO\_TogglePin**(GPIOD, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3);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

# BÀI 17 – CẢM BIẾN NHIỆT ĐỘ STM32

## Mô tả:

Ở bài này, yêu cầu

## Sơ đồ thiết kế:

****

## Đặc điểm linh kiện:

* DC Motor.
* Biến trở.

## Mã lệnh chính:

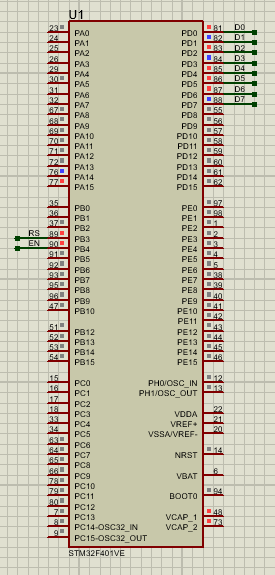
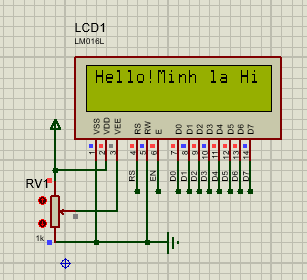
|  |
| --- |
| void **setup**()  {    }  void **loop**()  {    } |

# BÀI 18 – LCD STM32

## Mô tả:

Ở bài này, yêu cầu lắp màn hình LCD kích thước 16X2 vào mạch STM32F401VE và in ra màn hình LCD dòng chữ “Hello! Minh la Hieu”.

## Sơ đồ thiết kế:

## Đặc điểm linh kiện:

* 1 Mạch STM32F401VE.
* 1 Màn hình LCD kích thước 16X2.
* 1 Biến trở.

## Mã lệnh chính:

|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  //Ham gui cmd ra 8 chan cua LCD  void send8Bit2LCD(char D){  int b0, b1, b2, b3, b4, b5, b6, b7;  if((D & 1) == 0) b0=0;  else b0 = 1;  if((D & 2) == 0) b1=0;  else b1 = 1;  if((D & 4) == 0) b2=0;  else b2 = 1;  if((D & 8) == 0) b3=0;  else b3 = 1;  if((D & 16) == 0) b4=0;  else b4 = 1;  if((D & 32) == 0) b5=0;  else b5 = 1;  if((D & 64) == 0) b6=0;  else b6 = 1;  if((D & 128) == 0) b7=0;  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){  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  send8Bit2LCD(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 gui ky tu hien thi:  void sendChar2LCD(char Char){  HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_3, GPIO\_PIN\_SET);  send8Bit2LCD(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++)  sendChar2LCD(str[i]);  }  int main(void){  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  //Xoa noi dung LCD  sendCMD2LCD(0x01);  //Bat hien thi man hinh, tat con tro  sendCMD2LCD(0x0C);  //In chuoi ra LCD  sendString2LCD("Hello!Minh la Hieu");  while (1)  {  }  } |