

**Sinh viên : Nguyễn Lê Thành Tâm**

**Mssv : 61134311 Lớp : 61CNTT-1**

**HỌC PHẦN: LẬP TRÌNH NHÚNG**

**GIẢNG VIÊN : MAI CƯỜNG THỌ**

**NỘI DUNG**

**BÁO CÁO BÀI TẬP THỰC HÀNH LẬP TRÌNH ARDUINO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

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

**MỤC LỤC**

[**I. TỔNG QUAN LẬP TRÌNH NHÚNG**](#_Toc57630210)

[1. Arduino là gì ? 3](#_Toc57630229)

[2. Ứng dụng Arduino](#_Toc57630229) 3

[**II. BÀI TẬP**](#_Toc57630210)

[1. Nháy đèn LED trên cổng 13 3](#_Toc57630211)

[2. Điều](#_Toc57630212) khiển đèn LED bằng nút bấm 5

[3. Điểu khiển LED bằng cảm biến nhiệt độ](#_Toc57630213) 7

[4. LED RGB 9](#_Toc57630229)

[5. Điều khiển LED qua chiết áp 12](#_Toc57630229)

[6. Điều khiển hiển thị 00-99 bằng 2 LED 7 đoạn 14](#_Toc57630229)

[7. Điều khiển hiển thị LED 16](#_Toc57630229)

[8. Mô phỏng cảm biến khoảng cách HC-SR04 20](#_Toc57630229)

[9. Mô phỏng module L298 băm xung điều khiển động cơ motor 23](#_Toc57630229)

[10. Điều khiển động cơ Servo 2](#_Toc57630229)6

[11. Bật tắt đèn LED bằng STM32F401VE](#_Toc57630229) 29

[12. Lặp trình ngắt ngoài đơn giản bằng STM32F401VE](#_Toc57630229) 36

[13. Lập trình Tiner cơ bản với STM32](#_Toc57630229) 39

[14. Led metrix 4x4 với STM32F401VE](#_Toc57630229) 42

[11. Bật tắt đèn LED bằng STM32F401VE 2](#_Toc57630229)8

**III.** [**TÀI LIỆU THAM KHẢO**: 26](#_Toc57630232)

# I LÝ THUYẾT

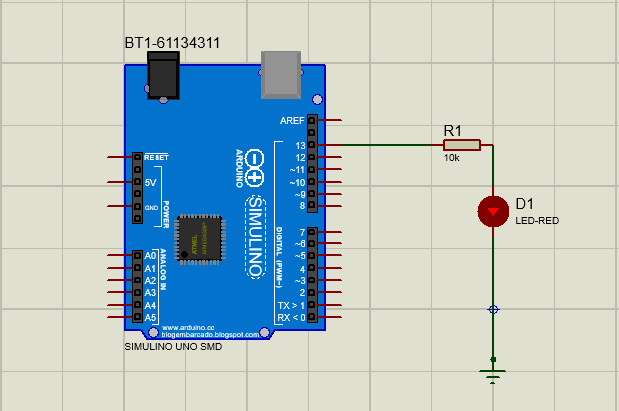
# II BÀI TẬP

## NHÁY LED TRÊN CỔNG 13

### Mô tả hệ thống

Hệ thống được thiết kế để thực hiện nháy đèn LED, đèn được nối vào Arduino ở cổng số 13, khoảng thời gian là 1000ms

### Sơ đồ thiết kế



### Đặc điểm linh kiện

* Arduino Uno board
* Điện trợ 10k Ω
* Đèn LED

### Mã lệnh

void setup() {.

pinMode(13, OUTPUT);

}

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 level

delay(1000);

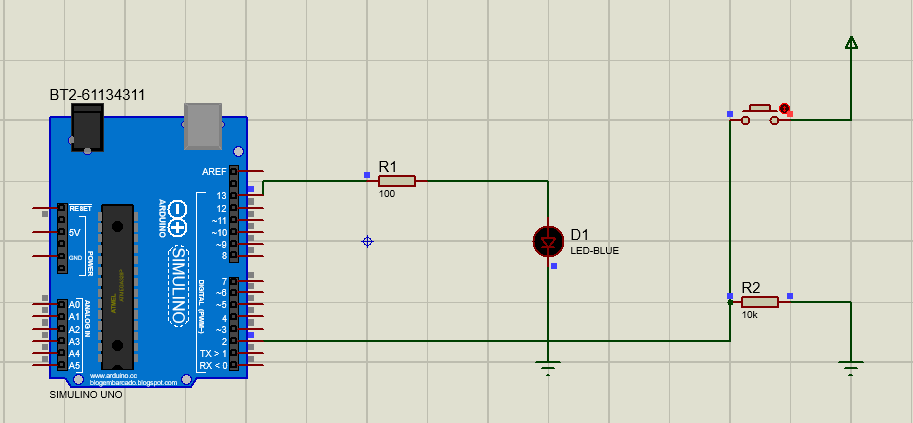
}

## ĐIỀU KHIỂN LED BẰNG NÚT BẤM

### Mô tả hệ thống

Hệ thống được cài đặt để thực hiện bật đèn LED thông qua nút bấm, LED được nối vào cổng 13 của Arduino, đầu ra nút bấm được nối với pin 2, nhấn nút bấm đèn sáng và ngược lại

### Sơ đồ thiết kế



### Đặc điểm linh kiện

* Arduino Uno board
* 1 điện trợ 10k Ω, 1 điện trở 100 Ω
* Đèn LED
* Nút bấm

### Mã lệnh

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);

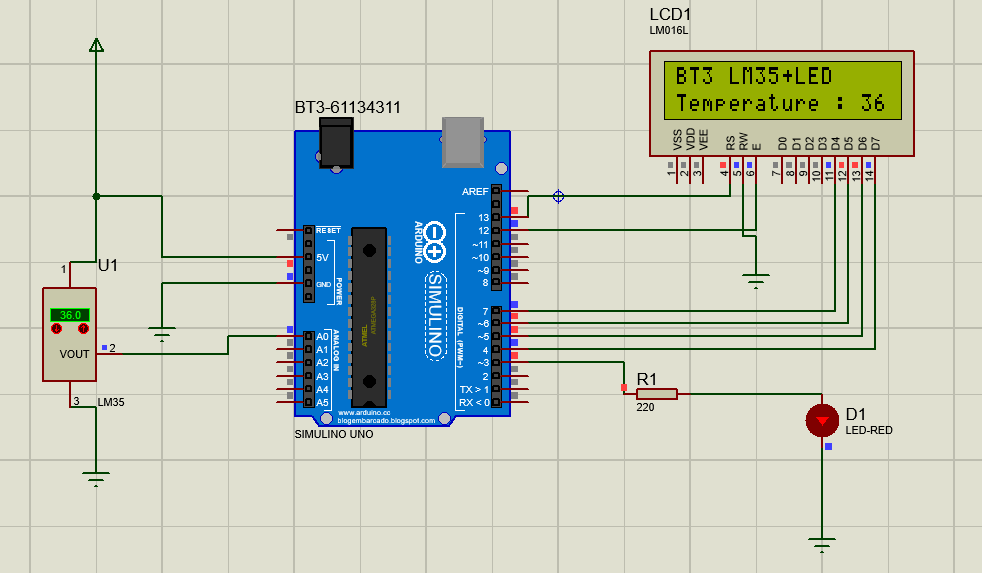
}

## ĐIỀU KHIỂN LED BẰNG CẢM BIẾN NHIỆT ĐỘ

### Mô tả hệ thống

Hệ thống được thiết lập để đọc nhiệt độ môi trường thông qua cảm biến TMP36, đèn LED sẽ sáng khi nhiệt độ trên 35 độ. LED được nối vào cổng 3 , đầu vào dữ liệu từ TMP36 từ cổng A0, nhiệt độ do được sẽ hiển thị trên màn hình LCD LM016L được nối vào mạch qua các cổng 12, 13

### Sơ đồ thiết kế



### Đặc điểm linh kiện

* + Đèn LED
  + Arduino Uno board
  + Cảm biến TMP36
  + Màn hình LCD
  + Điện trở 220 Ω

### Mã lệnh

#include <LiquidCrystal.h> // thêm thư viện cho màn hình LCD

LiquidCrystal lcd (13,12,7,6,5,4); // các cổng vào của LCD với mạch

#define led 3

void setup() {

pinMode(led, OUTPUT);

lcd.begin(16,2);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("BT3 LM35+LED");

delay(1000);

}

void loop() {

int value = analogRead(A0);// đọc tín hiệu analog từ TMP36

int temp = map(value, 0, 1023, 0, 500); // qui đổi sang nhiệt độ

lcd.setCursor(0,1);

lcd.print("Temperature : ");

lcd.print(temp);

if(temp > 35){

digitalWrite(led,1);}

else{

digitalWrite(led,0);}

delay(100);

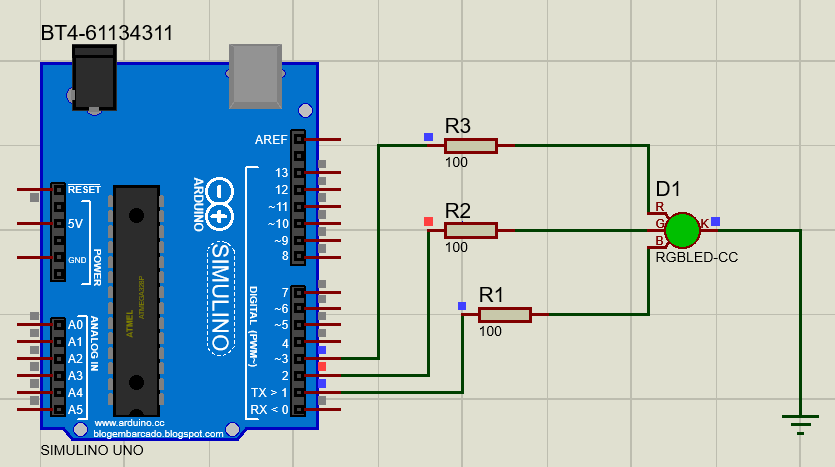
}

## ĐIỀU KHIỂN LED RGB

### Mô tả hệ thống

Hệ thống được thiết kế để hiển thị các màu sắc thông qua đèn LED RGB, Để thay đổi độ sáng của một con LED ta chỉ việc điều chỉnh điện áp xuất ra con LED, mà để điều chỉnh điện áp xuất ra con LED ta sẽ dùng xung PWM. 3 đầu của con LED được đấu nối lần lượt với các cổng 1, 2, 3 của Arduino thông qua 3 con trở để hạn dòng

### Sơ đồ thiết kế



### Đặc điểm linh kiện

* Arduino uno board
* LED RGB
* 3 điệ trở 100 Ω

### Mã lệnh

const int black = 0;

const int red = 1;

const int green = 2;

const int blue = 3;

const int yellow = 4;

const int cyan = 5;

const int magenta = 6;

const int white = 7;

void setup() {

pinMode(3,OUTPUT);

pinMode(2,OUTPUT);

pinMode(1,OUTPUT);

}

void loop() {

for (int i = black; i<= white;i++)

{

display (i);delay(600);

}

}

void display(int color){

switch(color){

case black : digitalWrite(1,0);

digitalWrite(2,0);

digitalWrite(3,0); break;

case red : digitalWrite(1,0);

digitalWrite(2,0);

digitalWrite(3 ,1); break;

case green : digitalWrite(1,0);

digitalWrite(2,1);

digitalWrite(3 ,0); break;

case blue : digitalWrite(1,0);

digitalWrite(2,1);

digitalWrite(3 ,1); break;

case yellow : digitalWrite(1,1);

digitalWrite(2,0);

digitalWrite(3 ,0); break;

case cyan : digitalWrite(1,1);

digitalWrite(2,0);

digitalWrite(3 ,1); break;

case magenta : digitalWrite(1,1);

digitalWrite(2,1);

digitalWrite(3 ,0); break;

case white : digitalWrite(1,1);

digitalWrite(2,1);

digitalWrite(3 ,1); break;

}

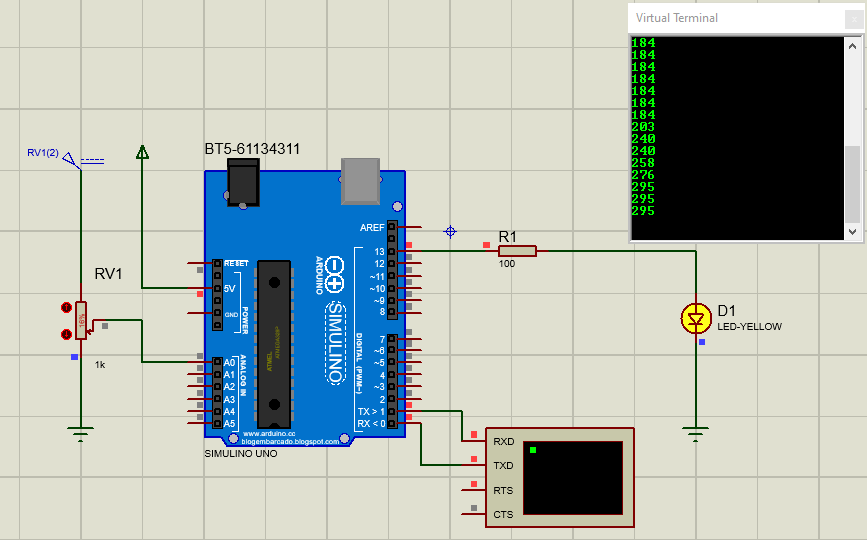
}

## ĐIỀU KHIỂN LED QUA CHIẾT ÁP

### Mô tả hệ thống

Hệ thống được lắp đặt để điều khiển đèn LED nhấp nháy theo một khoảng thời gian nhất định phụ thuộc vào sự thay đổi của chiếc áp, nếu giá trị điện áp ở cổng A0 càng cao thì đèn sẽ nhấp nháy trong khoảng thời gian càng lâu và ngược lại

### Sơ đồ thiết kế



### Đặc điểm linh kiện

* Arduino Uno board
* LED 0.1mA
* Điện trở 100 Ω
* Chiết áp POT-HG

### Mã lệnh

void setup() {

pinMode(A0,INPUT);

pinMode(13, OUTPUT);

Serial.begin(9600); // Khởi động 1 cổng Serial có baurate = 9600

}

int readA0 = 0;

void loop() {

readA0 = analogRead(A0); // mức điện áp ở chân A0 0->1023

Serial.println(readA0);

digitalWrite(13,HIGH);

delay(readA0);

digitalWrite(13,LOW);

delay(readA0);

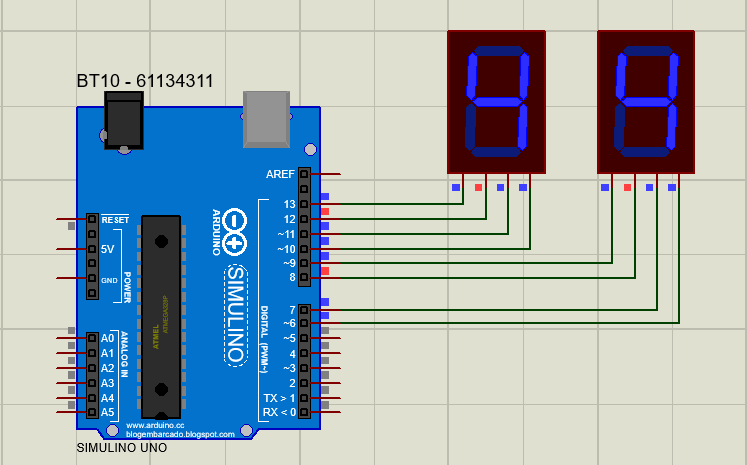
}

## ĐIỀU KHIỂN HIỂN THỊ 00 – 99 BẰNG 2 LED 7 ĐOẠN

### Mô tả hệ thống

Hệ thống được thiết lập để hiển thị trên 2 LED từ 00 đến 99, thông qua 2 thiết bị 7 Segment Binary Coded Decimal BCD và kết nối lần lượt với mạch Arduino qua các cổng 6 đến cổng 13

### Sơ đồ kết nối



### Đặc điểm linh kiện

* Arduino Uno board
* 2 LED 7 segment BCD

### Mã lệnh

void setup() {

pinMode(13,OUTPUT);pinMode(12,OUTPUT);pinMode(11,OUTPUT);

pinMode(10,OUTPUT);pinMode(9, OUTPUT);

pinMode(8, OUTPUT);pinMode(7, OUTPUT);pinMode(6, OUTPUT);

}

void loop() {

int a, b, c, d, e, f, g, h;

int x = 0; int y = 0;

for(a = 0; a < 2 & x<= 9; a++){

for(b = 0; b < 2 & x <= 9; b++){

for(c = 0; c < 2 & x <= 9; c++){

for(d = 0; d < 2 & x <= 9; d++){

for(e = 0; e < 2 & y <= 9; e++){

for(f = 0; f < 2 & y <= 9; f++){

for(g = 0; g < 2 & y <= 9; g++){

for(h = 0; h < 2 & y <= 9; h++){

digitalWrite(13, a);digitalWrite(12, b);digitalWrite(11, c);

digitalWrite(10, d);digitalWrite(9, e);digitalWrite(8, f);

digitalWrite(7, g);digitalWrite(6, h);

delay(300);

y = y+1;

}}}}

x = x+1;y = 0;}}}}

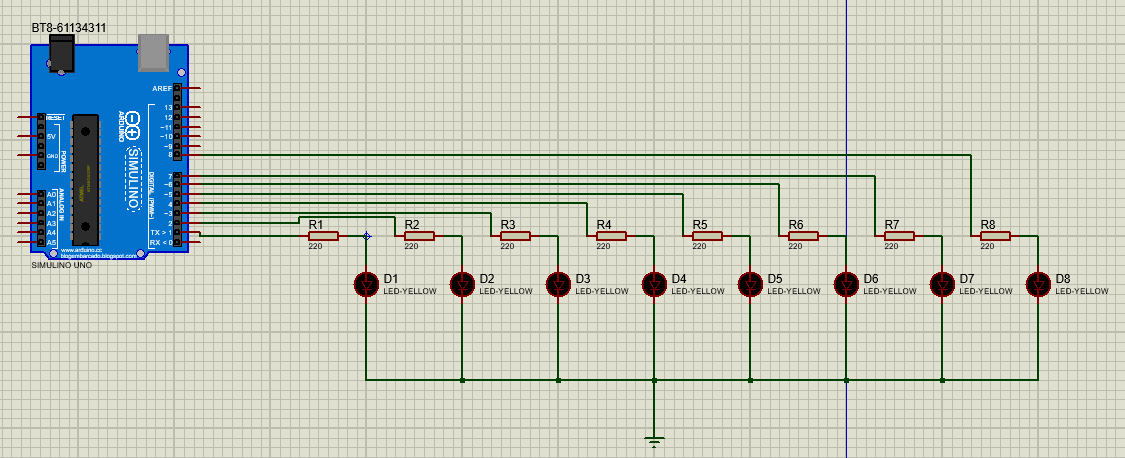
}

## ĐIỀU KHIỂN HIỂN THỊ 8 LED

### Mô tả hệ thống

Hệ thống được thiết lập để điều khiển 8 LED sáng nhấp nháy 8 , sáng dần từ trái sang và sáng dần từ phải sang trái, sáng xen kẽ và kết nối lần lượt với mạch Arduino qua các cổng 1 đến cổng 8

### Sơ đồ kết nối



### Đặc điểm linh kiện

* Arduino Uno board
* 8 đèn LED
* 8 điện trở 220 Ω

### Mã lệnh

byte ledPin[] = {1,2,3,4,5,6,7,8}; // mảng lưu vị trí các chân Digital mà các đèn LED sử dụng

byte pinCount; // pinCount dùng lưu tổng số chân LED

void setup() {

pinCount = sizeof(ledPin);

for (int i=0;i<pinCount;i++) {

pinMode(ledPin[i],OUTPUT); //các chân LED là OUTPUT

digitalWrite(ledPin[i],LOW); //mặc định các đèn LEDtắt

}

}

void TraiSangPhai()

{ //Bật lần lượt các đèn LED

for (int i=0; i < pinCount; i++) {

digitalWrite(ledPin[i],HIGH); //Bật đèn

delay(150);

}

}

void PhaiSangTrai()

{

for (int i=pinCount; i >=0; i--) {

digitalWrite(ledPin[i],HIGH); //Bật đèn

delay(150);

}

}

void TatLed()

{

for (int i=0;i<pinCount;i++) {

digitalWrite(ledPin[i],LOW); //mặc định các đèn LEDtắt

}

}

void TatLedLanLuot()

{

//Tắt lần lượt các đèn LED

for (int i = 0;i < pinCount; i++) {

digitalWrite(ledPin[i],LOW); // Tắt đèn

delay(150);

}

}

void SangLedCongChan()

{

TatLed();

digitalWrite(ledPin[2],HIGH); //Bật đèn ở cổng chẵn

digitalWrite(ledPin[4],HIGH); //Bật đèn ở cổng chẵn

digitalWrite(ledPin[6],HIGH); //Bật đèn ở cổng chẵn

digitalWrite(ledPin[8],HIGH); //Bật đèn ở cổng chẵn

delay(300);

}

void SangLedCongLe()

{

TatLed();

digitalWrite(ledPin[1],HIGH); //Bật đèn ở cổng chẵn

digitalWrite(ledPin[3],HIGH); //Bật đèn ở cổng chẵn

digitalWrite(ledPin[5],HIGH); //Bật đèn ở cổng chẵn

digitalWrite(ledPin[7],HIGH); //Bật đèn ở cổng chẵn

delay(300);

}

void NhapNhay()

{

TatLed();

for(int i = 0;i<=2;i++)

{

for (int i=0;i<pinCount;i++) {

digitalWrite(ledPin[i],LOW); //mặc định các đèn LEDtắt

}

delay(300);

for (int i=0;i<pinCount;i++) {

digitalWrite(ledPin[i],HIGH); //mặc định các đèn LEDtắt

}

delay(300);

}

void loop() {

TraiSangPhai();

TatLed();

PhaiSangTrai();

SangLedCongChan();

SangLedCongLe();

NhapNhay();

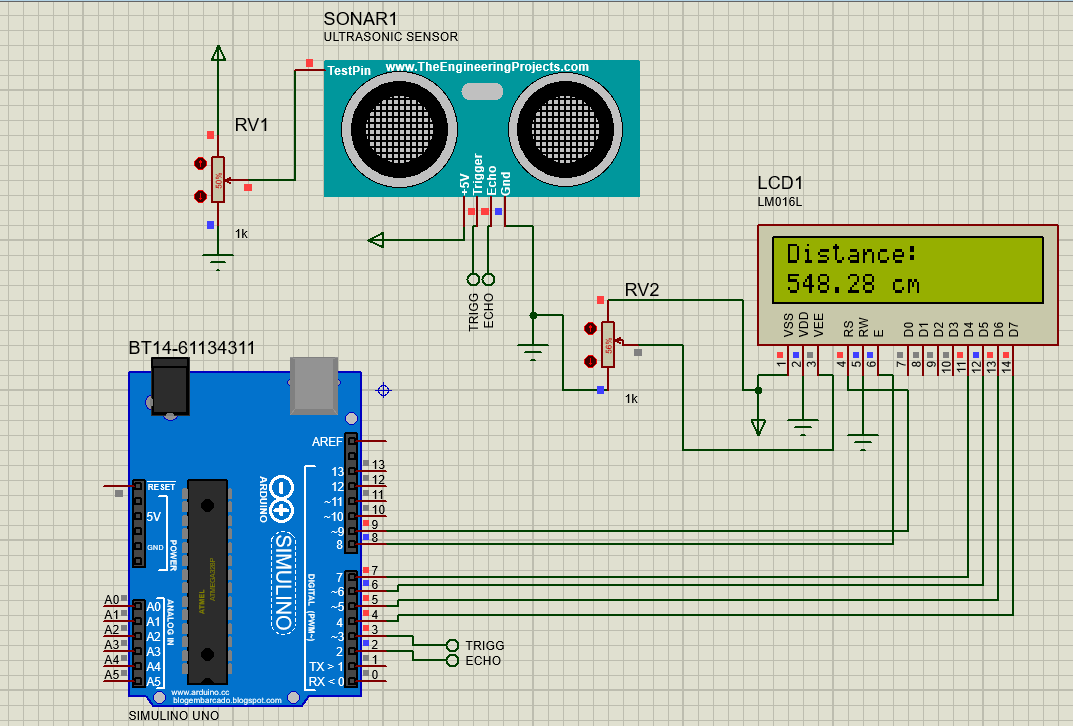
}

## MÔ PHỎNG CẢM BIẾN KHOẢNG CÁCH HC-SR04 VÀ HIỂN THỊ LÊN LCD

### Mô tả hệ thống

Hệ thống được thiết lập để mô phỏng cảm biến đo khoảng cách HC-SR04 và hiển thị giá trị độ rộng xung lên LCD, chân trigger dùng để phát sóng, chân echo là chân nhận sóng. Cảm biến sẽ phát sóng siêu âm đến vật cần đo, gặp vật cản, phản xạ lại cảm biến

### Sơ đồ kết nối



### Đặc điểm linh kiện

* Arduino Uno board
* Cảm biến HC-SR04,
* 2 biến trở POT-HG
* Màn hình LCD

### Mã lệnh

#include <LiquidCrystal.h> // thêm thư viện cho màn hình LCD

LiquidCrystal lcd (9,8,7,6,5,4); // các cổng vào của LCD với mạch

const int TRIGG = 3;

const int ECHO = 2;

void setup()

{

pinMode(TRIGG, OUTPUT);

pinMode(ECHO, OUTPUT);

lcd.begin(16,2);

}

void loop()

{

float duration;

float distance;

digitalWrite(TRIGG, 0);

delayMicroseconds(2);

digitalWrite(TRIGG, 1); // phat xung tu chan trigg

delayMicroseconds(10);

digitalWrite(TRIGG, 1);

duration = pulseIn(ECHO, HIGH,50000); /// do rong xung high sau 50000ms khong nhan duoc xung phan xa thi tra ve 0

distance = (duration \* 0.034/2);

lcd.setCursor(0, 0);

lcd.print("Distance:");

lcd.setCursor(0, 1);

lcd.print(distance);

lcd.print(" cm");

delay(1000);

lcd.clear();

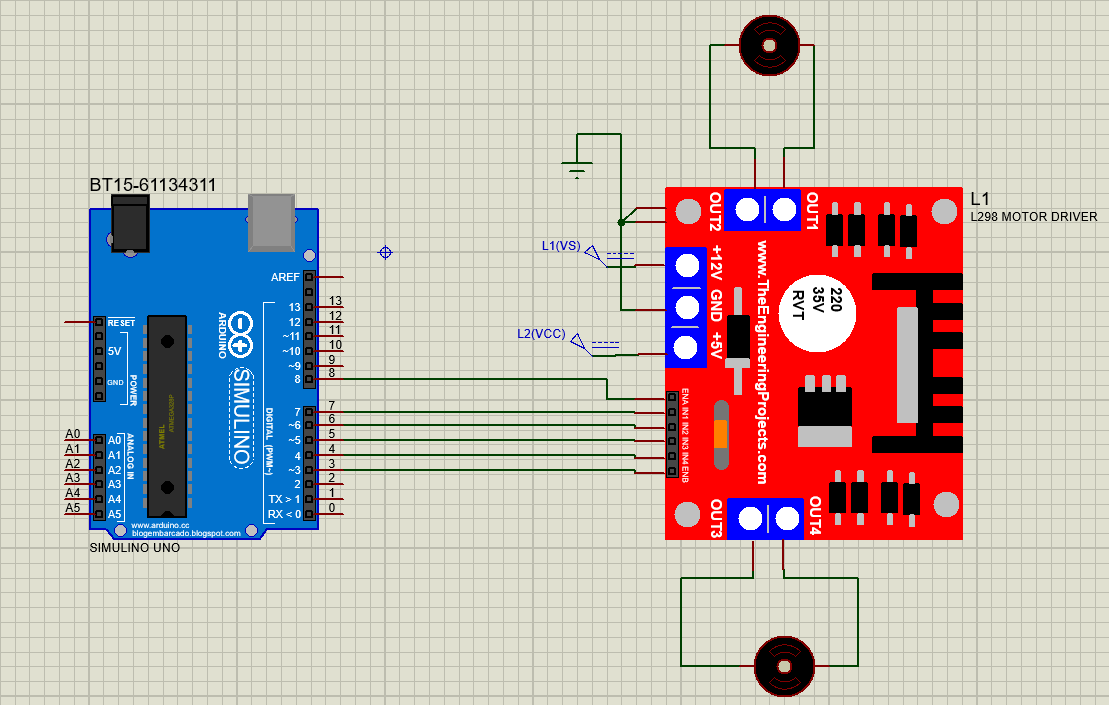
}

## MÔ PHỎNG MODULE L298 BĂM XUNG ĐIỀU KHIỂN MOTOR

### Mô tả hệ thống

Hệ thống được thiết lập để mô phỏng module điều khiển tốc độ motor tăng tốc, giảm tốc. Kết nối với mạch Arduino qua lần lượt các cổng từ 3 đến 7,2 chân ENA, ENB của module L298 thiết kế để điều khiển tốc độ động cơ

### Sơ đồ kết nối



### Đặc điểm linh kiện

* Arduino Uno board
* Module L298 motor driver,
* 2 động cơ DC

### Mã lệnh

// khai bao chan tin hieu motor A

int enA = 8;int in1 = 7;int in2 = 6;

// khai bao chan tin hieu motor B

int enB = 3;int in3 = 5;int in4 = 4;

int i;

void setup() {

pinMode(enA,OUTPUT);

pinMode(in1,OUTPUT);

pinMode(in2,OUTPUT);

pinMode(enB,OUTPUT);

pinMode(in3,OUTPUT);

pinMode(in4,OUTPUT);

}

void runmotor()

{

// tang toc

for (i = 0;i<=255;i++)

{

digitalWrite(in3,1);

digitalWrite(in1,1);

digitalWrite(in4,0);

digitalWrite(in2,0);

analogWrite(enB,i);

analogWrite(enA,i);

delay(100);

// giam toc

for (i = 255;i>=0;i--)

{

digitalWrite(in3,1);

digitalWrite(in1,1);

digitalWrite(in4,0);

digitalWrite(in2,0);

analogWrite(enB,i);

analogWrite(enA,i);

delay(100);}

}

void loop() {

runmotor();

delay(1000);

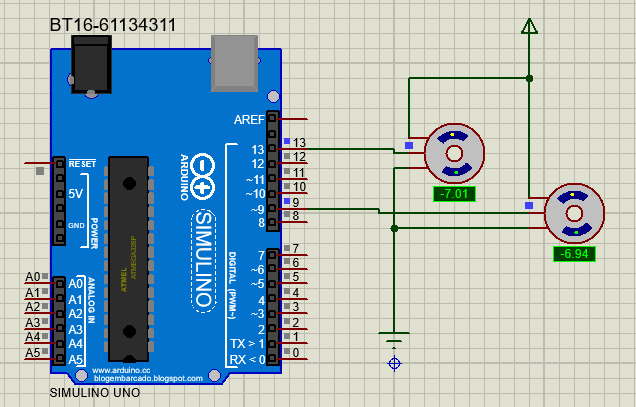
}

## ĐIỀU KHIỂN ĐỘNG CƠ SERVO

### Mô tả hệ thống

Hệ thống được thiết lập để mô phỏng điều khiển động cơ Servo quay 180 độ cùng và ngược chiều kim đồng hồ thông qua việc kết nối với cổng 9 và 13 của mạch Arduino,

### Sơ đồ kết nối



### Đặc điểm linh kiện

* Arduino Uno board
* 2 động cơ motor Servo

### Mã lệnh

#include <Servo.h>

Servo s;

Servo s1;

int pos = 0;

void setup() {

s.attach(9);

s1.attach(13);

}

void loop() {

for(pos = 0; pos <= 180;pos++)

{

s.write(pos);

s1.write(pos);

delay(10);

}

for (pos =180;pos>=0;pos--)

{

s.write(pos);

s1.write(pos);

delay(10);

}

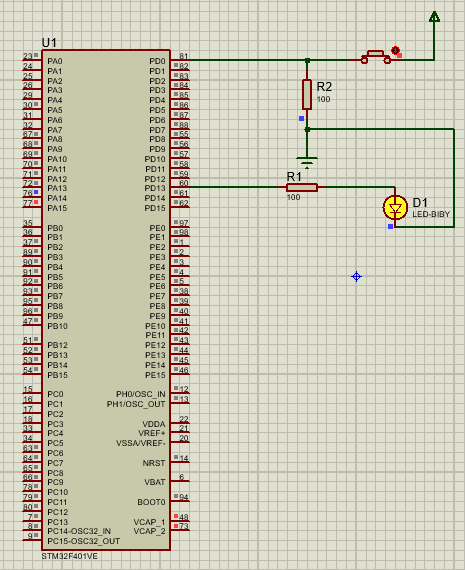
}

## BẬT TẮT ĐÈN LED BẰNG STM32F401VE

### Mô tả hệ thống

Hệ thống được thiết lập để mô phỏng điều khiển bật tắt đèn LED thông qua nút bấm, với viêc thiết lập cổng D0 là INPUT, nhận trạng thái của nút bấm và ghi giá trị ra cổng OUTPUT D13 để điều khiển bóng đè sáng

### Sơ đồ kết nối



### Đặc điểm linh kiện

* Chip STM32F401VE
* 1 đèn LED,
* 1 Điện trở 100 Ω

### Mã lệnh

int main(void)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

while (1)

{

GPIO\_PinState PIN0 = HAL\_GPIO\_ReadPin(GPIOD,GPIO\_PIN\_0);

if(PIN0 == GPIO\_PIN\_SET)

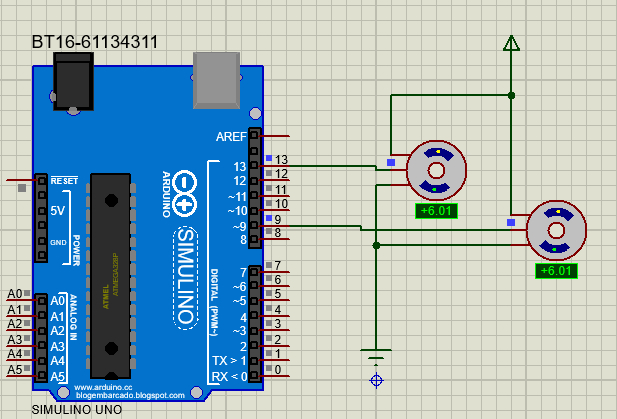
HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_13,GPIO\_PIN\_SET);

else

HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_13,GPIO\_PIN\_RESET);

}

}

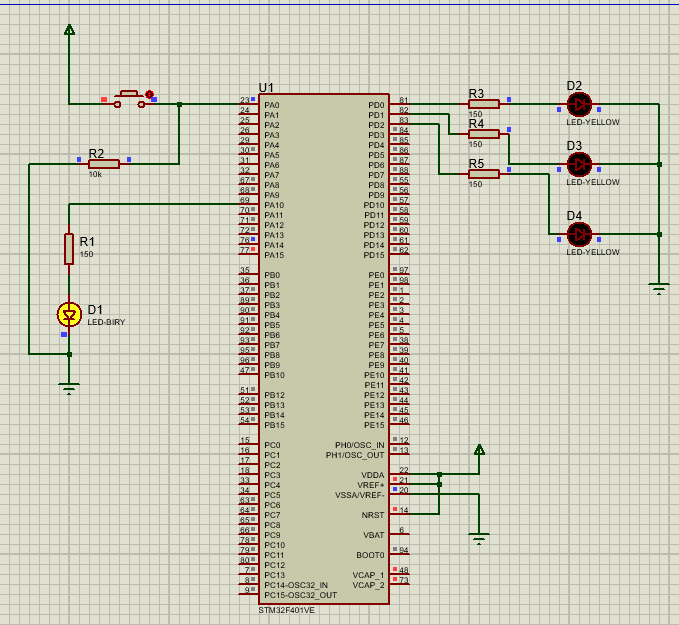


## LẬP TRÌNH NGẮT NGOÀI ĐƠN GIẢN

### Mô tả hệ thống

Hệ thống được thiết lập để mô phỏng điều khiển bật tắt đền LED thông qua nút bấm. Ngắt ngoài và ưu tiên ngắt trên STM32F4, đèn D1 được nối với chân A10, thông tin nút bấm được ghi thông qua chân A0, khi được nhận thì sáng đèn D1, việc nhấp nháy các đèn D2, D3, D4 vẫn được thực hiện dù xảy ra ngắt

### Sơ đồ kết nối



### Đặc điểm linh kiện

* Chip STM32F401VE
* 4 đèn LED,
* 5 Điện trở 150 Ω

### Mã lệnh

int main(void)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

while (1)

{

HAL\_GPIO\_TogglePin(GPIOD,GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2);

HAL\_Delay(1000);

}

}

void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)

{

if(GPIO\_PIN\_0==GPIO\_PIN\_0)

{ HAL\_GPIO\_TogglePin(GPIOA,GPIO\_PIN\_10);

}

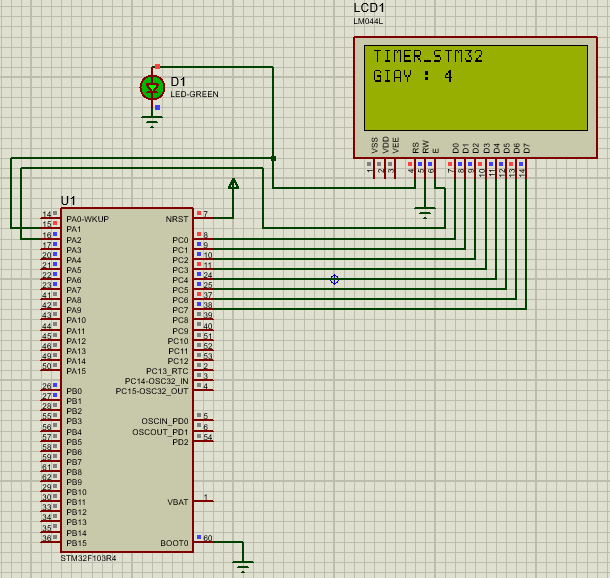
}

## LẬP TRÌNH TIMER CƠ BẢN VỚI STM32

### Mô tả hệ thống

Hệ thống được thiết lập để thực hiện hiện hiển thị thời gian theo giây lên màn hình LCD khi kết nói được với Timer, đồng thời đó là dùng Timer để điều khiển một LED sáng. Với FCK\_PSC=8MHz = 8.000.000 Hz; ta có thể chọn PSC=999, và ARR = 7999

### Sơ đồ kết nối



### Đặc diểm linh kiện

* Chip STM32F103R4
* 1 đèn LED,
* Màn hình LCD LM LM044L

### Mã lệnh

void lcd\_command(char ml)

{

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_1,GPIO\_PIN\_RESET);

GPIOC->ODR = ml;

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_2,GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_2,GPIO\_PIN\_RESET);

HAL\_Delay(1);

}

void lcd\_loaddata(char ml)

{

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_1,GPIO\_PIN\_SET);

GPIOC->ODR = ml;

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_2,GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_2,GPIO\_PIN\_RESET);

HAL\_Delay(1);

}

void lcd\_string(const char \*s){

while(\*s) lcd\_loaddata(\*s++) ;

}

void lcd\_init(){

lcd\_command(0x38);HAL\_Delay(5);

lcd\_command(0x38);HAL\_Delay(5);

lcd\_command(0x0c);

lcd\_command(0x01);HAL\_Delay(2);

}

void HAL\_TIM\_PeriodElapsedCallback(TIM\_HandleTypeDef \*htim){

if(htim->Instance == TIM1){

giay++;

HAL\_GPIO\_TogglePin(GPIOA,GPIO\_PIN\_1);

}

}

int main(void){

HAL\_Init();

SystemClock\_Config();MX\_GPIO\_Init();MX\_TIM1\_Init();

lcd\_init();

lcd\_string("TIMER\_STM32");

HAL\_TIM\_Base\_Start\_IT(&htim1);

while (1)

{ sprintf(ht,"GIAY : %d",giay);

lcd\_command(0xc0);

lcd\_string(ht);

}

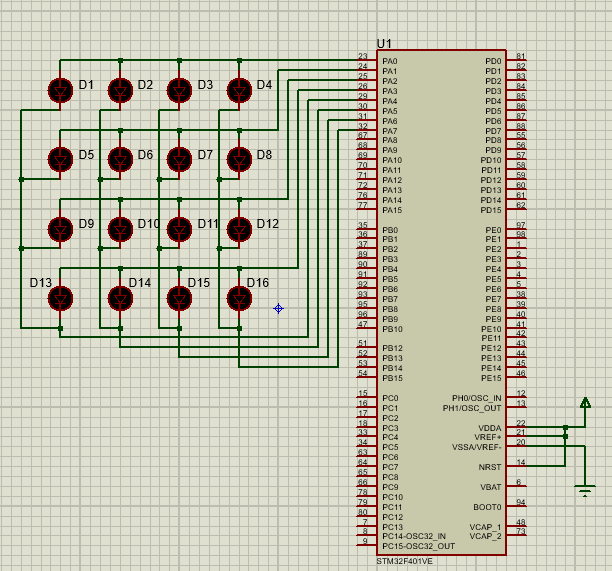
}

## LED METRIX 4X4 VỚI STM32F401VE

### Mô tả hệ thống

Hệ thống được thiết lập để thực hiện hiện hiển thị một số hiệu ứng trên ma trận LED 4x4 thông qua chip STM32F4VE

### Sơ đồ kết nối



### Đặc điểm linh kiện

* Chip STM32F103R4
* 16 đèn LED

### Mã lệnh

#include "main.h"

void SystemClock\_Config(void);

static void MX\_GPIO\_Init(void);

void effect1()

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_0, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, GPIO\_PIN\_RESET);

HAL\_Delay(100);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5, GPIO\_PIN\_RESET);

HAL\_Delay(100);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, GPIO\_PIN\_RESET);

HAL\_Delay(100);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, GPIO\_PIN\_RESET);

HAL\_Delay(100);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_0, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, GPIO\_PIN\_SET);

HAL\_Delay(100);

}

int main(void)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

/\* USER CODE BEGIN 2 \*/

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7, GPIO\_PIN\_SET);

while (1)

{

effect2();

HAL\_Delay(250);

effect1();

}

}

int main(void)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7, GPIO\_PIN\_SET);

while (1)

{

efect1();

HAL\_Delay(250);

//effect2();

}

}