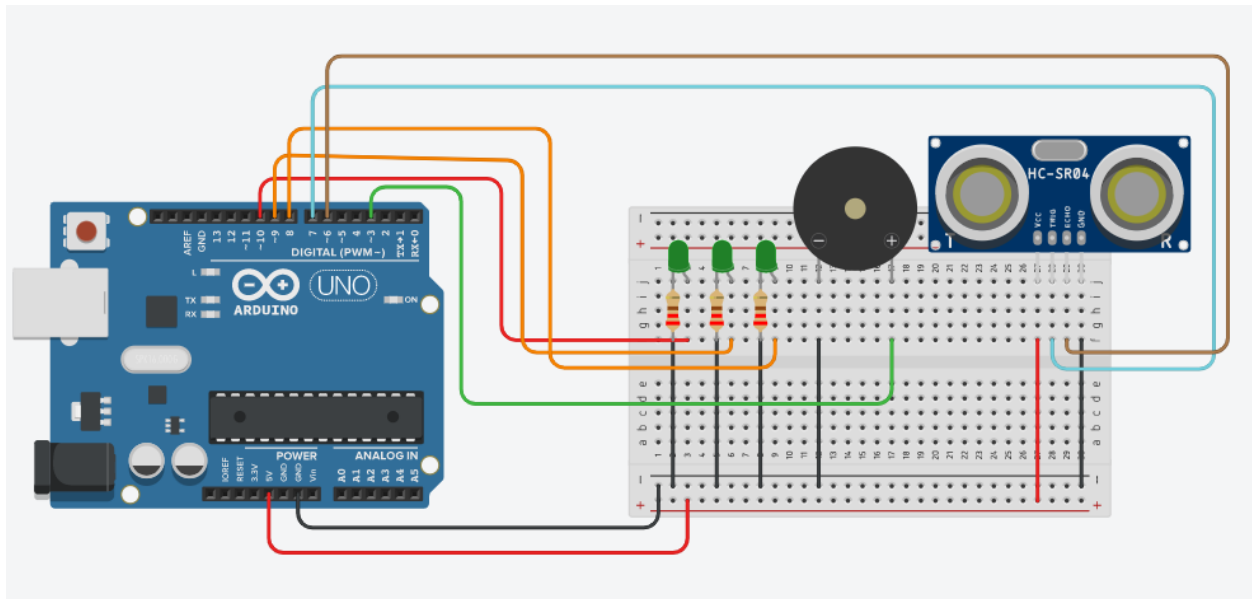


MỤC LỤC

Câu 1: HC-SR04	2
Câu 2: Hiện thị nhiệt độ trên LCD.....	7
Câu 3: Gas sensor – cảm biến khí gas	10
Câu 4: Nhấn nút sáng đèn – pushbutton	13
Câu 5: Relay	16
Câu 6: Mạch điều khiển truy cập mật khẩu Arduino Tinkercad với LCD	17
Câu 6: Điều khiển motor qua hệ thống hồng ngoại	28
Câu 7: Điều khiển servo bằng button.....	30
Câu 8: Điều khiển 8 led.....	32
Câu 9: Điều khiển motor bằng H-bridge Motor Driver	35
Câu 10: Điều khiển motor và loa bằng hệ thống hồng ngoại	37
Câu 11: Điều khiển đèn led thông qua giao tiếp UART	39
Câu 12: Led giao thông	40

Câu 1: HC-SR04

Sơ đồ mạch điện



Hình 1: Sơ đồ đầu nối mạch điện đo khoảng cách.

Mô tả sơ đồ hoạt động

- Khoảng cách < 100cm, 3 đèn sáng cùng lúc.
- Khoảng cách < 200cm, đèn 1 tắt – đèn 2, 3 sáng.
- Khoảng cách < 300cm, đèn 1, 2 tắt – đèn 3 sáng.
- Khoảng cách < 120cm, loa sẽ hú cảnh báo.

Linh kiện mạch

- HC-SR04
- LED
- Loa piezo
- Resistor 110Ω
- Arduino Uno R3

Mô tả chi tiết linh kiện

HC-SR04

HC-SR04 sử dụng để xác định khoảng cách . Cảm biến sử dụng sóng siêu âm và có thể đo khoảng cách trong khoảng từ 2 - > 300 cm.



LED

Đèn led là ứng dụng của hiện tượng phát quang của chip led.



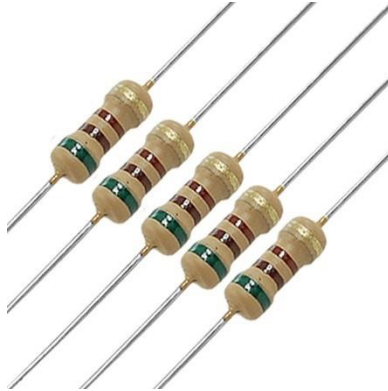
Loa piezo

Loa piezo là loại loa phát âm sử dụng hiệu ứng áp điện để chuyển đổi tín hiệu điện ra âm thanh.



Resistor 220Ω

Resistor chức năng của nó dùng để điều chỉnh mức độ tín hiệu, hạn chế cường độ dòng điện chảy trong mạch. Dùng để chia điện áp, kích hoạt các linh kiện điện tử



Arduino Uno R3

Arduino Uno R3 đi kèm với giao diện USB, 6 chân đầu vào analog, 14 cổng kỹ thuật số I / O được sử dụng để kết nối với các mạch điện tử, thiết bị bên ngoài. Trong đó có 14 cổng I / O, 6 chân đầu ra xung PWM cho phép các nhà thiết kế kiểm soát và điều khiển các thiết bị mạch điện tử ngoại vi một cách trực quan.



Code

```
#define trigPin 7
#define echoPin 6
#define led4 10
#define led5 9
#define led6 8
#define buzzer 3

int sound = 250;

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
  pinMode(led6, OUTPUT);
  pinMode(buzzer, OUTPUT);
}

void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

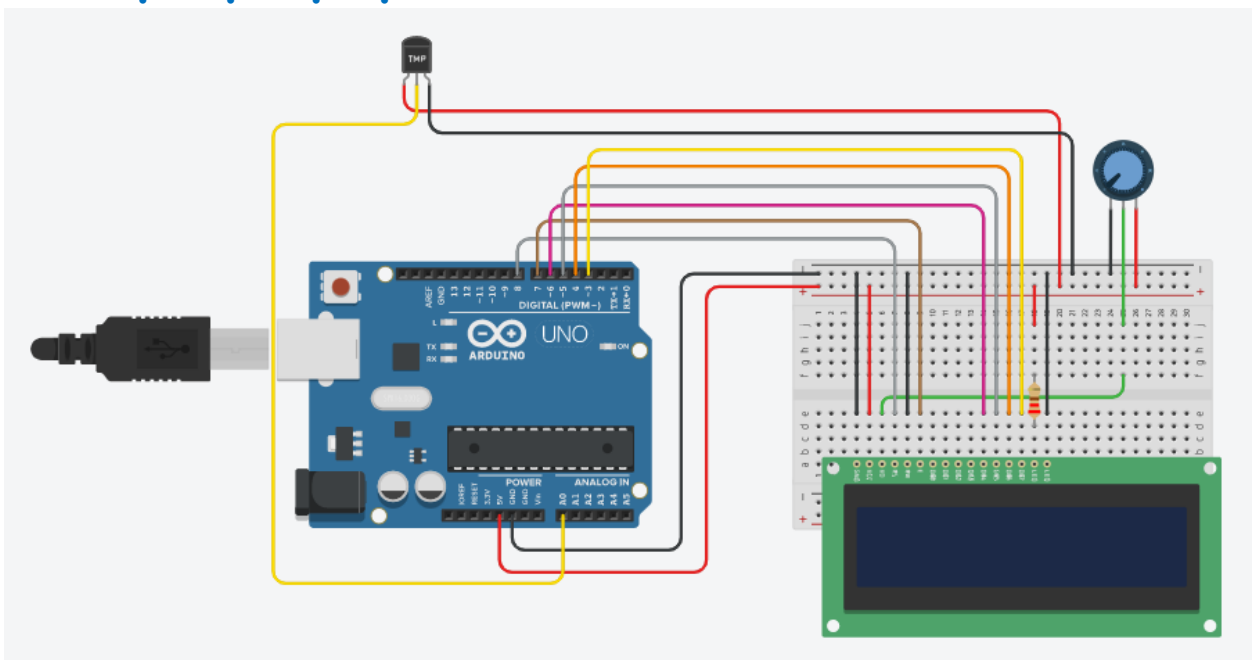
```
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
if (distance < 100) {
  digitalWrite(led4, HIGH);
  sound = 280;
}
else {
  digitalWrite(led4, LOW);
}
if (distance < 200) {
  digitalWrite(led5, HIGH);
  sound = 290;
}
else {
  digitalWrite(led5, LOW);
}
if (distance < 300) {
  digitalWrite(led6, HIGH);
  sound = 300;
}
else {
  digitalWrite(led6, LOW);
}
if (distance > 120 || distance <= 0){
  Serial.print("Out of range");
```

```

noTone(buzzer);
}
else {
Serial.print(distance);
Serial.println(" cm");
tone(buzzer, sound);
}
delay(500);
}

```

Câu 2: Hiện thị nhiệt độ trên LCD



Hình 2: Sơ đồ đấu nối mạch điện hiện thị nhiệt độ.

Mô tả sơ đồ hoạt động

- Di chuyển thanh nhiệt độ temperature sensor tmp36 để hiện thị nhiệt độ.

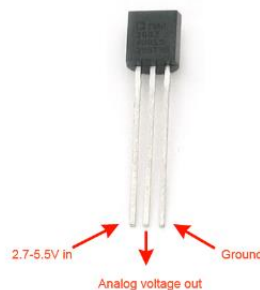
Linh kiện mạch

- Temperature sensor [TMP36]
- LCD 16*2
- Resistor 110Ω
- Arduino Uno R3

Mô tả chi tiết linh kiện

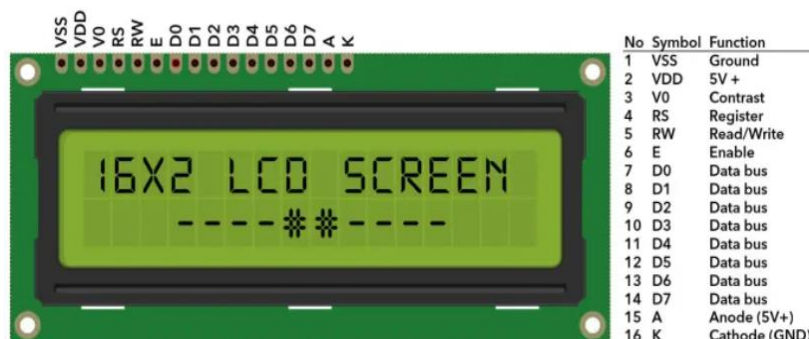
Temperature sensor [TMP36]

TMP36 là cảm biến nhiệt độ điện áp thấp, chính xác. Nó cung cấp một đầu ra điện áp tỷ lệ tuyến tính với nhiệt độ C. Nó cũng không yêu cầu bất kỳ hiệu chuẩn bên ngoài nào để cung cấp độ chính xác điển hình là $\pm 1^\circ\text{C}$ ở $+25^\circ\text{C}$ và $\pm 2^\circ\text{C}$ trong phạm vi nhiệt độ -40°C đến $+125^\circ\text{C}$.



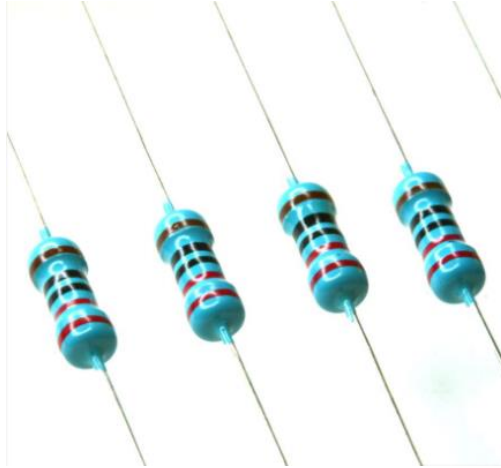
*LCD 16*2*

LCD 16x2 được sử dụng để hiển thị trạng thái hoặc các thông số. LCD 16x2 có 16 chân trong đó 8 chân dữ liệu (D0 - D7) và 3 chân điều khiển (RS, RW, EN). 5 chân còn lại dùng để cấp nguồn và đèn nền cho LCD 16x2. Các chân điều khiển giúp ta dễ dàng cấu hình LCD ở chế độ lệnh hoặc chế độ dữ liệu.



Resistor 220Ω

Resistor chức năng của nó dùng để điều chỉnh mức độ tín hiệu, hạn chế cường độ dòng điện chảy trong mạch. Dùng để chia điện áp, kích hoạt các linh kiện điện tử



Code

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,7,6,5,4,3);
int sensorPin = 0;

void setup()
{
  Serial.begin(9600);
  lcd.begin(16,2);
}
void loop()
{
  int i;
  int reading = analogRead(sensorPin);
  float voltage = reading * 4.68;
```

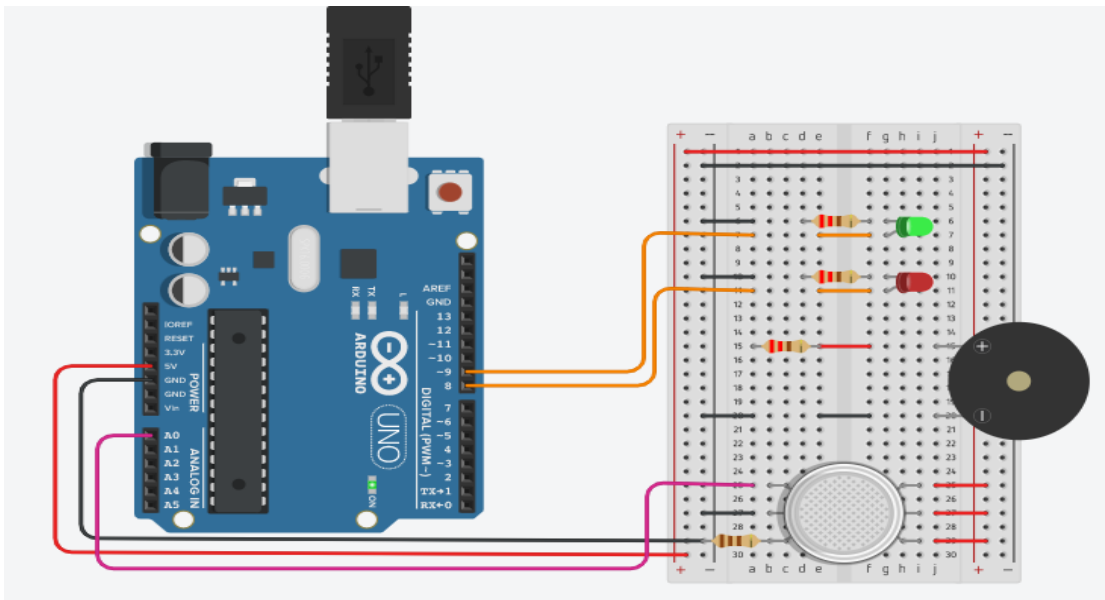
```

voltage /= 1024.0;
float temperatureC = (voltage - 0.5) * 100;
Serial.print(temperatureC);
Serial.println(" degrees C");
lcd.setCursor(i,0);
lcd.print(" Temperature Value ");
lcd.setCursor(0,1);
lcd.print(" deg. C");
lcd.setCursor(10,1);
lcd.print(temperatureC);
delay(100);
}

```

Câu 3: Gas sensor – cảm biến khí gas

Sơ đồ mạch điện



Hình 3: Sơ đồ đấu nối mạch điện gas sensor – cảm biến khí gas

Mô tả sơ đồ hoạt động mạch

- Khí Gas > 100, đèn đỏ sáng và chuông sẽ cảnh báo rò khí Gas.
- Khí Gas < 100, đèn xanh sáng nằm trong mức an toàn.

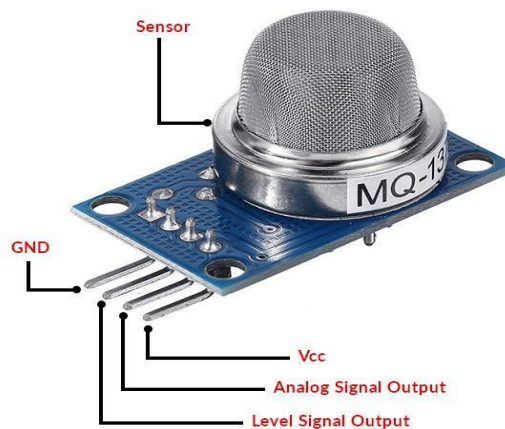
Linh kiện mạch

- LED
- GAS SENSOR
- Loa piezo
- Resistor 110Ω, 1kΩ.
- Arduino Uno R3

Mô tả chi tiết linh kiện mạch

GAS SENSOR

Gas sensor sử dụng các phản ứng vật lý hoặc hóa học để chuyển đổi nồng độ của các loại khí khác nhau thành tín hiệu điện và xuất ra các giá trị sau khi tính toán. Được sử dụng rộng rãi để phát hiện khí độc và có hại và rò rỉ khí tự nhiên.



Code

```
int ledro = 9;
int ledxanh = 8;
int loa = 6;
```

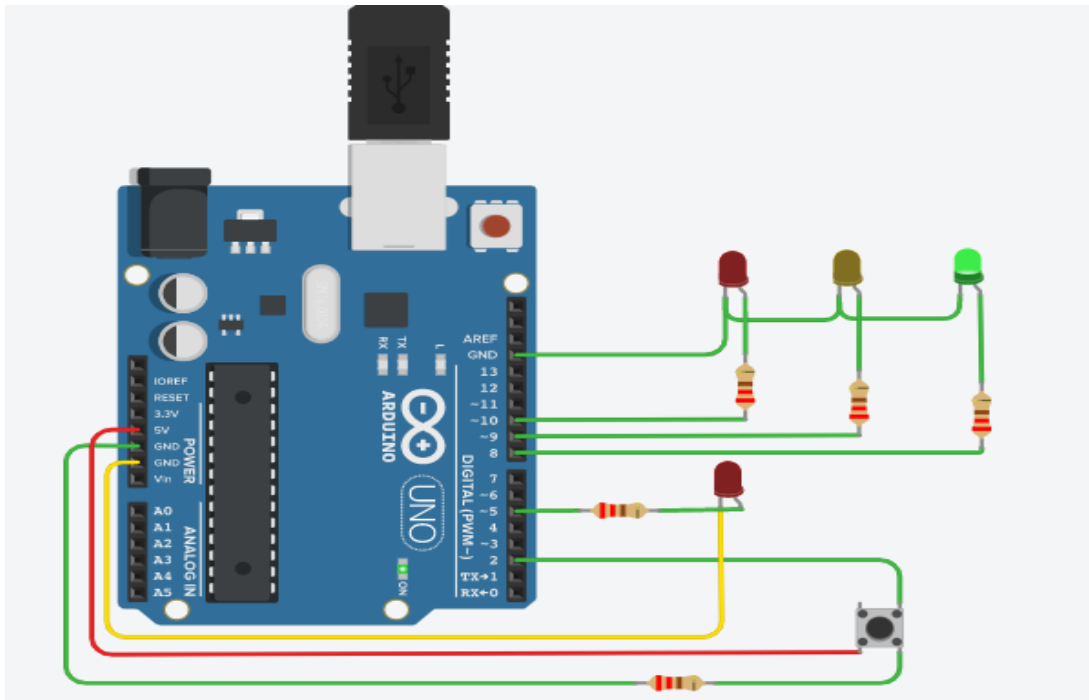
```
int cbgas = A1;

void setup() {
  pinMode(leddo, OUTPUT);
  pinMode(ledxanh, OUTPUT);
  pinMode(loa, OUTPUT);
  pinMode(cbgas, INPUT);
}

void loop() {
  int gtcb = analogRead(cbgas);
  Serial.println(gtcb);
  if ( gtcb > 150 )
  {
    digitalWrite(leddo, HIGH);
    digitalWrite(ledxanh, LOW);
    tone (loa,100,1000);
  }
  else {
    digitalWrite(leddo, LOW);
    digitalWrite(ledxanh, HIGH);
    noTone (loa);
  }
}
```

Câu 4: Nhấn nút sáng đèn – pushbutton

Sơ đồ mạch điện



Hình 4: Sơ đồ đấu nối mạch điện nhấn nút sáng đèn – pushbutton

Mô tả sơ đồ hoạt động mạch

- Nhấn giữ chuột trái vào nút -> đèn sáng.
- Không nhấn nút -> đèn tắt.

Linh kiện mạch

- LED
- PUSHBUTTON
- Resistor 220Ω
- Arduino Uno R3

Mô tả chi tiết linh kiện mạch

PUSHBUTTON

Push button có nghĩa là Nút nhấn.



Code

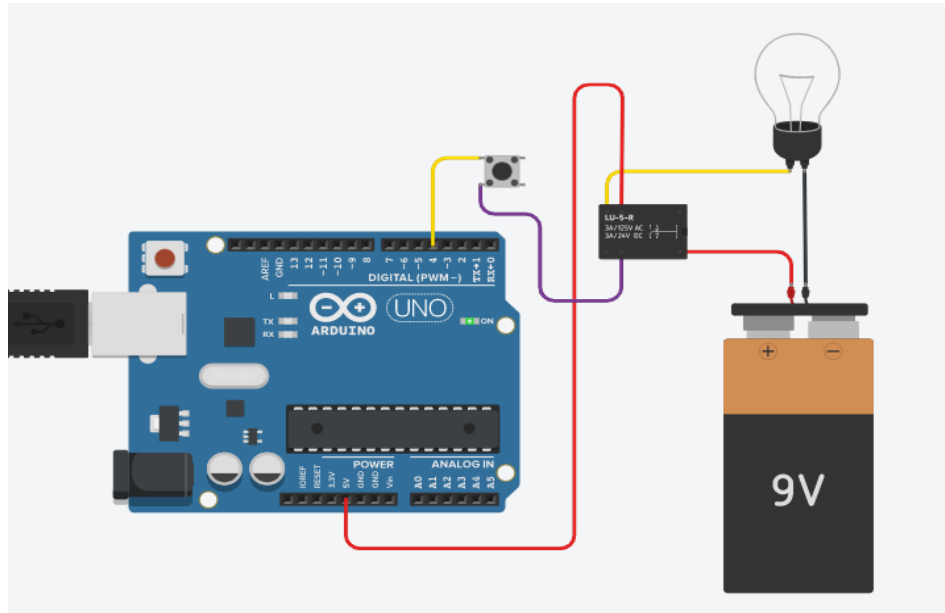
```
int R = 10;
int G = 9;
int Y = 8;
int denNutBam = 5;
void setup()
{
  pinMode(R, OUTPUT);
  pinMode(G, OUTPUT);
  pinMode(Y, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(2, INPUT);
  pinMode(denNutBam, OUTPUT);
  attachInterrupt(0, NgatSangLed, CHANGE);
}
void loop()
{
  //led giao thong
  digitalWrite(R, HIGH); // den do bat
  delay(10000); // )
  digitalWrite(R, LOW);
```

```
    delay(100);
    digitalWrite(G, HIGH); // den xanh bat
    delay(3000); //
    digitalWrite(G, LOW);
    delay(100);
    digitalWrite(Y, HIGH); // den vang bat
    delay(10000);
    digitalWrite(Y, LOW);
    delay(100);
}

void NgatSangLed()
{
    int trangThaiNut = digitalRead(2);
    Serial.print(trangThaiNut);
    if (trangThaiNut == HIGH)
    {
        digitalWrite(denNutBam, HIGH);
        delay(1000);
    }
    else
    {
        digitalWrite(denNutBam, LOW);
        delay(100);
    }
}
```

Câu 5: Relay

Sơ đồ mạch điện



Hình 5: sơ đồ đấu nối mạch điện relay

Mô tả sơ đồ hoạt động mạch

- Khi nhấn nút điện chạy qua relay thì điện sáng với thời gian là 4s.

Linh kiện mạch

- Arduino Uno R3
- PUSHBUTTON
- Relay SPDT
- Light bulb
- 9V Battery

Mô tả chi tiết linh kiện mạch

RELAY SPDT

SPDT là tên gọi viết tắt của Single Pole Double Throw. Có nghĩa là: có hai tiếp điểm thường đóng (NC) và thường mở (NO) với một tiếp điểm COM (tiếp điểm chung – hay tiếp điểm giữa).

Chúng ta có 2 cặp tiếp điểm:

- Tiếp điểm C-B gọi là NC (Normal Close)
- Tiếp điểm C- A gọi là NO (Normal Open)



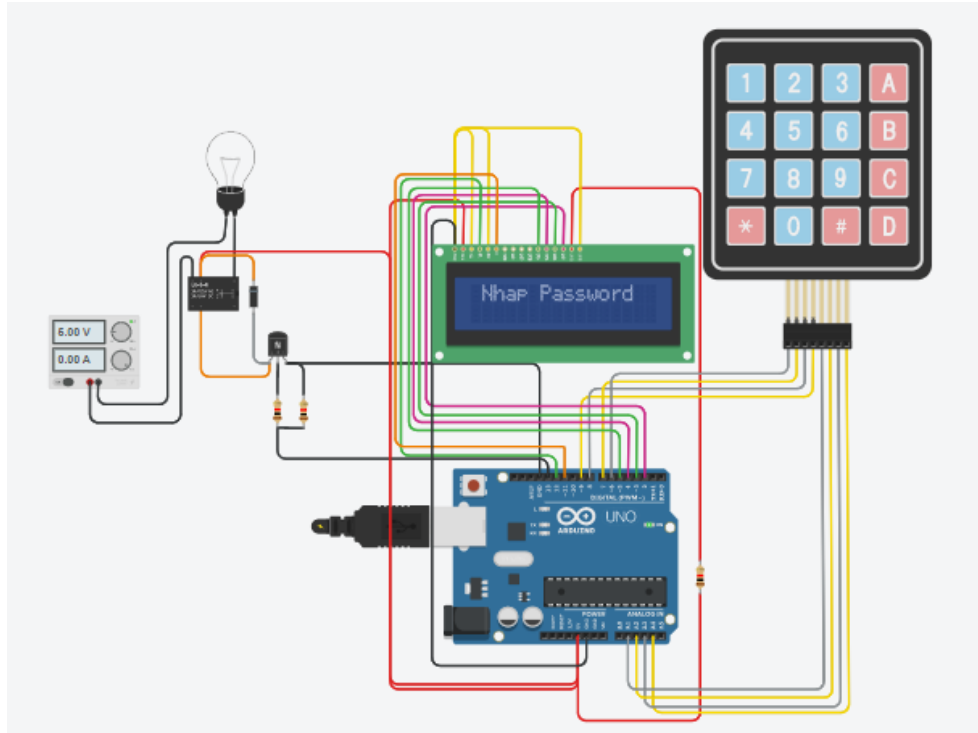
Code

```
void setup()
{
  pinMode(4, OUTPUT);
}

void loop()
{
  digitalWrite(4, HIGH);
  delay(4000); // Wait for 10000 millisecond(s)
  digitalWrite(4, LOW);
  delay(2000); // Wait for 2000 millisecond(s)
}
```

Câu 6: Mạch điều khiển truy cập mật khẩu Arduino Tinkercad với LCD

Sơ đồ mạch điện



Hình 6: Sơ đồ đấu nối mạch điện Arduino Tinkercad với LCD

Mô tả sơ đồ hoạt động mạch

- Khi nhập đúng mật khẩu thì đèn sẽ sáng lên và hiển thị mở khóa
- Khi nhập sai thì sẽ hiển thị sai mật khẩu và cho phép nhập lại

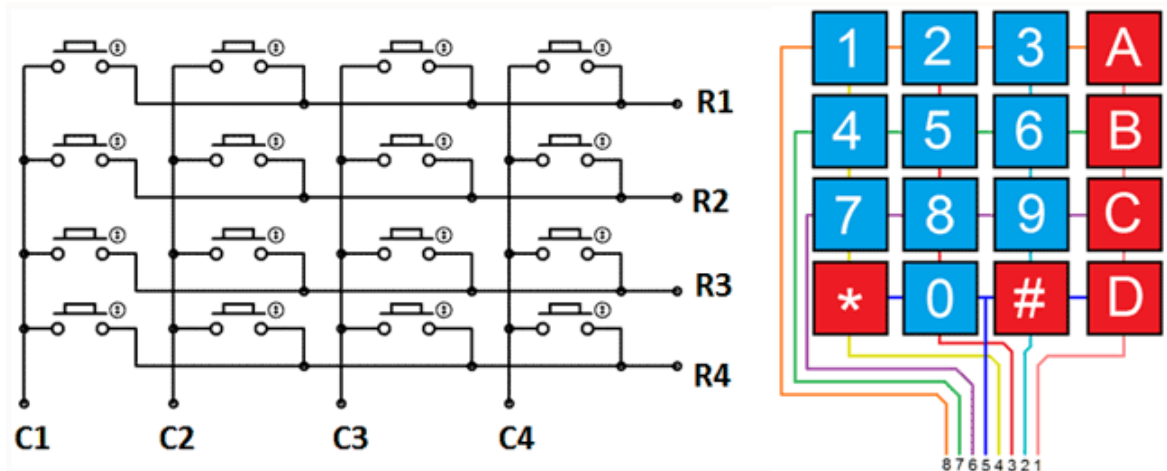
Linh kiện mạch

- Arduino Uno R3
- Keypad 4x4
- Lcd 16x2
- Relay SPDT
- NPN Transistor (BJT)
- Đèn
- 12, 5 Power Supply
- Diode

- Resistor

Mô tả chi tiết linh kiện mạch

- Keypad 4x4



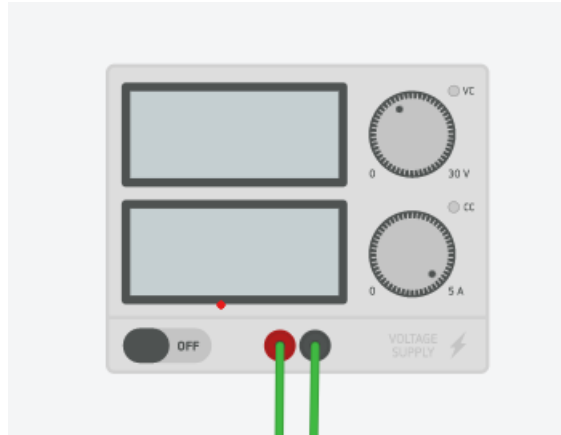
Một bàn phím là một tập hợp các nút hoặc phím mang chữ số, biểu tượng và / hoặc chữ cái đặt theo thứ tự trên một pad, có thể được sử dụng như một thiết bị đầu vào hiệu quả. Một bàn phím có thể hoàn toàn số, như tìm thấy trên một máy tính hoặc một khóa cửa kỹ thuật số, hoặc chữ số như những người sử dụng trên điện thoại di động.

- Relay SPDT



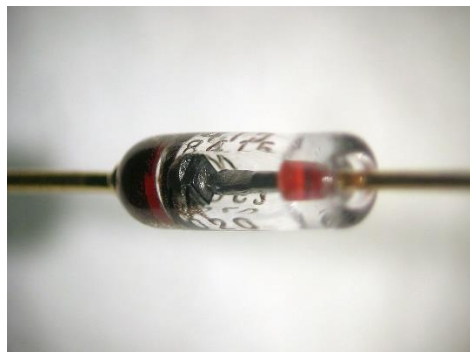
Một chuyển đổi-loại phổ biến là SPDT. SPDTs có ba nhà ga: pin một trong những phổ biến và hai chân mà vie cho kết nối để phổ biến. SPDTs là rất tốt cho việc lựa chọn giữa hai nguồn năng lượng, trao đổi các yếu tố đầu vào, hoặc bất cứ điều gì nó là bạn với hai mạch cố gắng để đi một nơi. Nút nhấn đặt kim loại thiết bị chuyển mạch là của nhiều loại SPDT. SPDT thiết bị chuyển mạch thường nên có ba thiết bị đầu cuối hoặc 5termianls bao gồm 2poles cho led chiếu sáng.

- Power Supply



Nguồn điện là một thiết bị điện cung cấp năng lượng điện cho tải điện. Mục đích chính của nguồn điện là chuyển đổi dòng điện từ nguồn sang điện áp, dòng điện và tần số chính xác để cấp nguồn cho tải. Do đó, bộ nguồn đôi khi được gọi là bộ chuyển đổi năng lượng điện. Một số bộ nguồn là các thiết bị độc lập riêng biệt, trong khi một số khác được tích hợp vào các thiết bị tải mà chúng cung cấp năng lượng. Ví dụ về cái sau bao gồm nguồn điện được tìm thấy trong máy tính để bàn và các thiết bị điện tử tiêu dùng

- Diode



Diode là một thành phần điện tử hai cực dẫn dòng điện chủ yếu theo một hướng (độ dẫn điện không đối xứng); Nó có mức kháng cự thấp (lý tưởng bằng không) theo một hướng và điện trở cao (lý tưởng là vô hạn) theo hướng khác.

Code

```
#include <LiquidCrystal.h>
```

```
#include <Keypad.h>
```

```
#include <EEPROM.h>

#define Password_Lenght 5

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};
byte rowPins[ROWS] = {6, 7, 8, 9};
byte colPins[COLS] = {A1, A2, A3, A4};

char Data[Password_Lenght];
char Data2[Password_Lenght];
char Master[Password_Lenght];
byte data_count = 0, master_count = 0;
bool Pass_is_good;
char key;
byte key_state = 0;
char last_press_key;
```

```
byte mode = 0;
```

```
int change_password_allow_time = 10000;
```

```
long time_old = 0;
```

```
bool just_allowed_pass = 0;
```

```
char FistTimePassword[] = {'1','2','3','4'}; // setup first-time password here
```

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS  
);
```

```
void setup(){
```

```
    Serial.begin(9600);
```

```
    Check_EEPROM();
```

```
    lcd.begin(16, 2);
```

```
    pinMode(13, OUTPUT);
```

```
    digitalWrite(13, LOW);
```

```
}
```

```
void loop()
```

```
{
```

```
    key = keypad.getKey();
```

```
key_state = keypad.getState();
```

```
if(key){  
    last_press_key = key;  
    Serial.println(key);  
}
```

```
if( mode == 3){  
    if(last_press_key == '#' && key_state == 2){  
        mode = 1;  
    }  
    if(last_press_key == '*' && key_state == 2){  
        mode = 0;  
        lcd.clear();  
        lcd.setCursor(4,0);  
        lcd.print("LOCKED");  
        digitalWrite(13, LOW);  
        delay(2000);  
    }  
}
```

```
if(mode == 0){  
    lcd.setCursor(1,0);  
    lcd.print("Nhap Password");
```

```
}else if(mode == 1){  
    lcd.setCursor(0,0);  
    lcd.print("Set New Password");  
}else if(mode == 2){  
    lcd.setCursor(0,0);  
    lcd.print("Password Again");  
}else if(mode == 3){  
    lcd.setCursor(4,0);  
    lcd.print("UNLOCKED");  
}
```

```
if (key && key != '#' && mode != 3)  
{  
    collectKey();  
}
```

```
if(data_count == Password_Lenght-1)  
{  
    if(mode == 0){  
        lcd.clear();  
        if(!strcmp(Data, Master)) {  
            lcd.setCursor(2, 0);
```



```
lcd.print("WELCOME BACK");  
lcd.setCursor(4, 1);  
lcd.print("MASTER");  
  
just_allowed_pass = 1;  
time_old = millis();  
  
digitalWrite(13, HIGH);  
  
delay(2000);  
mode = 3;  
}else{  
  lcd.setCursor(2, 0);  
  lcd.print("INCORRECT !");  
  lcd.setCursor(4, 1);  
  lcd.print("PASSWORD");  
  delay(2000);  
}  
delay(1000);//  
lcd.clear();  
clearData();  
  
// digitalWrite(13, LOW);  
}else if( mode == 1){  
  lcd.clear();
```

```

mode = 2;
for(int i = 0; i < Password_Lenght; i = i+1){
  Data2[i] = Data[i];
}
clearData();
}else if(mode == 2){
  if(!strcmp(Data, Data2)){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("New Password is ");
    lcd.setCursor(4, 1);
    lcd.print(Data);
    delay(2000);
    lcd.clear();
    lcd.setCursor(4, 0);
    lcd.print("Saving...");
    for(int i =0; i <= 100; i = i+10){
      lcd.setCursor(4, 1);
      lcd.print(i);
      lcd.setCursor(7, 1);
      lcd.print("%");
      delay(200);
    }
    EEPROM.put(0, Data);
    EEPROM.get(0, Master);
  }
}

```

```

        delay(500);
    }else{
        lcd.clear();
        lcd.setCursor(4, 0);
        lcd.print("PASSWORD");
        lcd.setCursor(3, 1);
        lcd.print("NOT MATCH!");
        delay(2000);
    }
    mode = 3;
    clearData();
    lcd.clear();
}
}
}

void collectKey(){
    Data[data_count] = key;
    lcd.setCursor(4+data_count,1);
    lcd.print("*");
    data_count++;
}

void clearData()
{
    while(data_count !=0)
    {

```

```

    Data[data_count--] = 0;
}
}

void Check_EEPROM(){
    EEPROM.get(0, Master);

    if(Master[0] == 0 && Master[1] == 0 && Master[2] == 0 && Master[3] == 0){
// check if EEPRM have store password ?

        Serial.println("No EEPROM PASSWORD FOUND"); // if not found will burn
        EEPROM a first time password

        EEPROM.put(0, FistTimePassword);

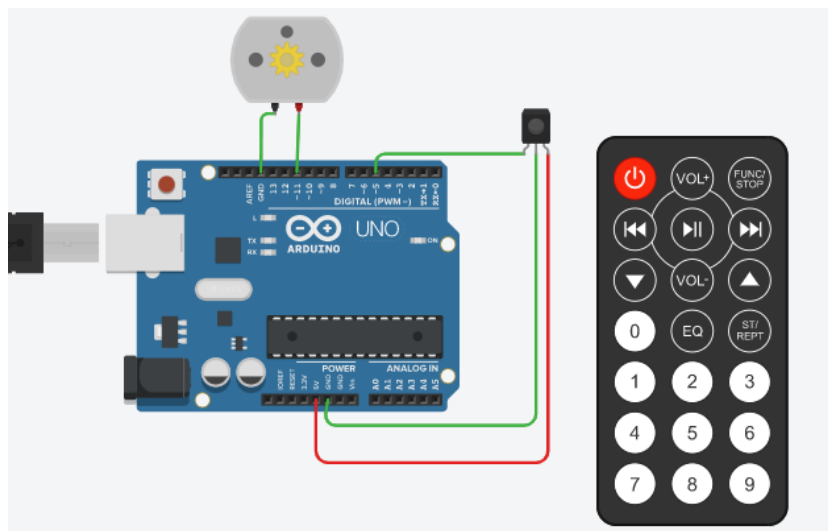
        EEPROM.get(0, Master);

    }
}

```

Câu 6: Điều khiển motor qua hệ thống hồng ngoại

Sơ đồ mạch điện



Hình 6: Sơ đồ đấu nối mạch điện điều khiển motor

Mô tả sơ đồ hoạt động mạch

- Động cơ motor quay khi nhấn điều khiển nút power, nhấn nút 1 động cơ tắt.

Linh kiện mạch

- Arduino Uno R3.
- DC Motor.
- IR sensor.
- IR remote.

Mô tả chi tiết linh kiện mạch

Code

```
#include <IRremote.h>

IRrecv matHongNgoai(5);
decode_results maNhan;

void setup()
{
  pinMode(11, OUTPUT);
  pinMode(5, INPUT);
  matHongNgoai.enableIRIn(); //Cho phép hoạt động ngay từ đầu
  Serial.begin(9600); // Cấu hình cổng Serial để in mã nhận
}

void loop()
{
  matHongNgoai.decode(&maNhan); //Tiến hành giải mã

  Serial.println(maNhan.value); //in mã để xem
  //Các xử lý với mã code ở đây
```

```

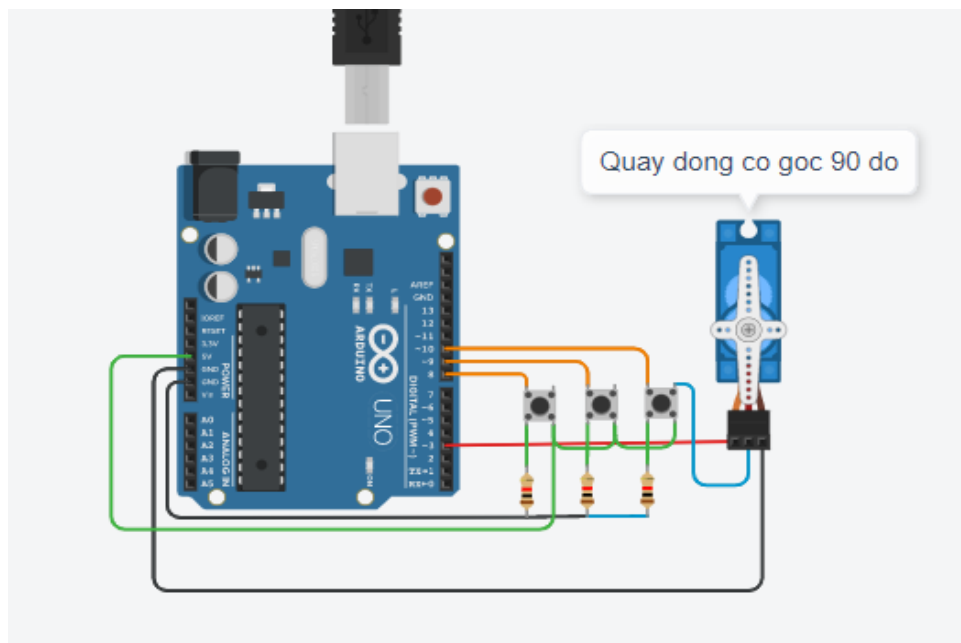
//Power code: 16580863
//Vol-: 16617583
//so 1: 16582903
//Kêu to nhất (255) --> nhỏ nhất (0)
if (maNhan.value == 16582903) //phím 1
    analogWrite(11, 255); //Quay nhanh nhất (255) --> chậm nhất (0)
else
    analogWrite(11, 0);
matHongNgoai.resume(); //Tiếp tục lắng nghe

delay(1000);
}

```

Câu 7: Điều khiển servo bằng button

Sơ đồ mạch điện



Hình 7: Sơ đồ đấu nối mạch điện điều khiển servo bằng button

Mô tả sơ đồ hoạt động mạch

- Nhấn nút A: Servo quay 0 độ.
- Nhấn nút B: Servo quay 45 độ.
- Nhấn nút C: Servo quay 90 độ.

Linh kiện mạch

- Arduino Uno R3.
- Pushbutton.
- Resistor.
- Micro Servo.

Mô tả chi tiết linh kiện mạch

Code

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo

//int potpin = A0; // analog pin used to connect the potentiometer
//int val; // variable to read the value from the analog pin

void setup()
{
  pinMode(8, INPUT);
  pinMode(9, INPUT);
  pinMode(10, INPUT);
  myservo.attach(3); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  int nutTT8 = digitalRead(8);
```

```

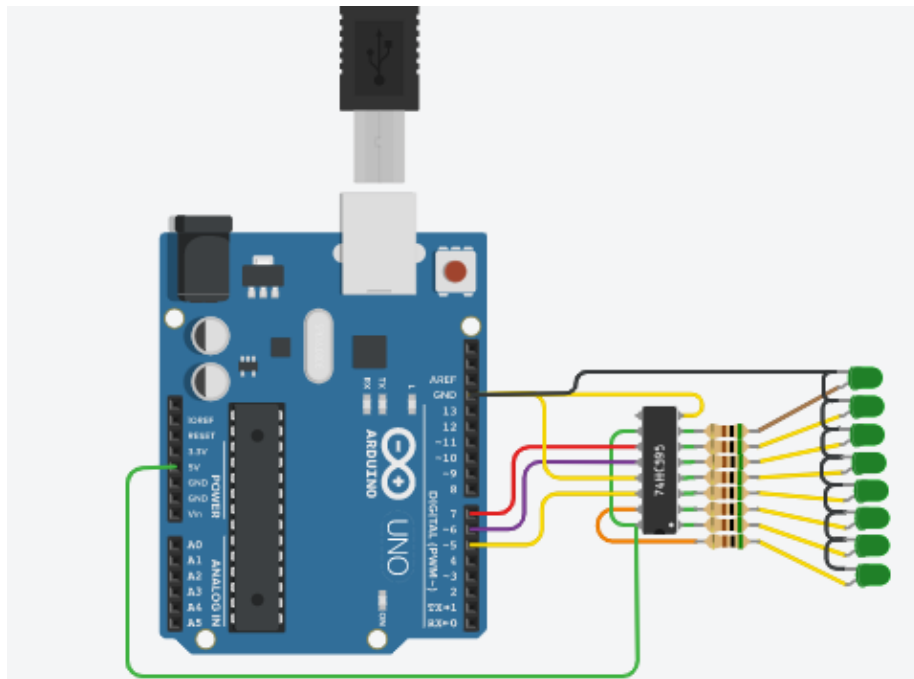
if(nutTT8 == HIGH)
    myservo.write(0);
int nutTT9 = digitalRead(9);
if(nutTT9 == HIGH)
    myservo.write(45);
int nutTT10 = digitalRead(10);
if(nutTT10 == HIGH)
    myservo.write(90);

delay(100); // waits for the servo to get there
}

```

Câu 8: Điều khiển 8 led

Sơ đồ mạch điện



Hình 8: Sơ đồ đấu nối mạch điện đèn led

Mô tả sơ đồ hoạt động mạch

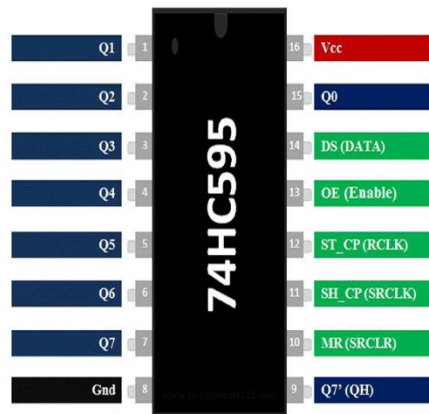
- Điều khiển hệ thống đèn led sáng tất cả và sáng xen kẽ từng đèn.

Linh kiện mạch

- Arduino Uno R3.
- 74HC595 8-Bit Shift Register.
- Resistor.
- Led.

Mô tả chi tiết linh kiện mạch

- 74HC595 là một thanh ghi dịch (shift register) hoạt động trên giao thức nối tiếp vào song song ra (Serial IN Parallel OUT). Nó nhận dữ liệu nối tiếp từ vi điều khiển và sau đó gửi dữ liệu này qua các chân song song. Có thể tăng 8 chân đầu ra bằng cách sử dụng chip đơn.



Code

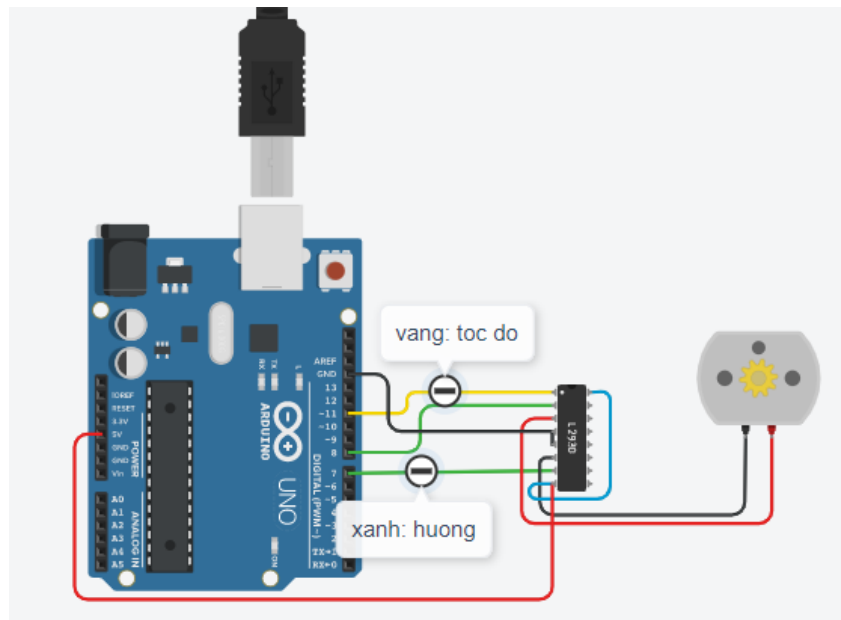
```
void setup()
{
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
}

void loop()
```

```
{  
    // Chot lai  
    digitalWrite(6, LOW);  
    delay(1000);  
    // shift data  
    shiftOut(5,7, LSBFIRST, 170);  
    digitalWrite(6, HIGH);  
    delay(1000);  
    digitalWrite(6, LOW);  
    delay(1000);  
    shiftOut(5,7, LSBFIRST, 0);  
    digitalWrite(6, HIGH);  
    delay(1000);  
    digitalWrite(6, LOW);  
    delay(1000);  
    shiftOut(5,7, LSBFIRST, 255);  
    digitalWrite(6, HIGH);  
    delay(1000);  
    digitalWrite(6, LOW);  
    delay(1000);  
    shiftOut(5,7, LSBFIRST, 0);  
    digitalWrite(6, HIGH);  
    delay(1000);  
}
```

Câu 9: Điều khiển motor bằng H-bridge Motor Driver

Sơ đồ mạch điện



Hình 9: Sơ đồ đấu nối mạch điện điều khiển motor bằng H-bridge motor driver

Mô tả sơ đồ hoạt động mạch

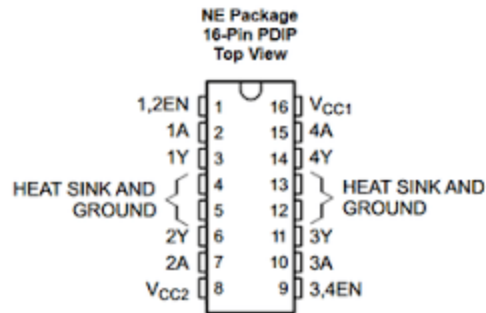
- Điều khiển motor quay theo chiều kim đồng hồ với tốc độ cao nhất trong 2 giây sau đó dừng 3 giây và quay ngược chiều kim đồng hồ trong 2 giây.

Linh kiện mạch

- Arduino Uno R3.
- L293D H-bridge Motor Driver.
- DC Motor.

Mô tả chi tiết linh kiện mạch

- L293D là IC cầu H điều khiển động cơ. Gồm 4 kênh điều khiển có thể điều khiển 2 động cơ DC hoặc 1 động cơ bước 4 pha (5 dây). Để điều khiển động cơ DC, bạn sẽ sử dụng 2 kênh của L293D cho 1 động cơ. L293D đã được tích hợp sẵn đi ốt bảo vệ vì điều khiển chống lại dòng cảm ứng khi động cơ khởi động hoặc tắt.



Code

```
// 11 toc do
// 8 huong A
// 7 huong B

void setup()
{
    pinMode(11, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(7, OUTPUT);
}

void loop()
{
    /*
    // quay chieu kim dong ho
    digitalWrite(8, HIGH);
    digitalWrite(7, LOW);
    // toc do
    analogWrite(11, 255);
    delay(5000);
    */
}
```

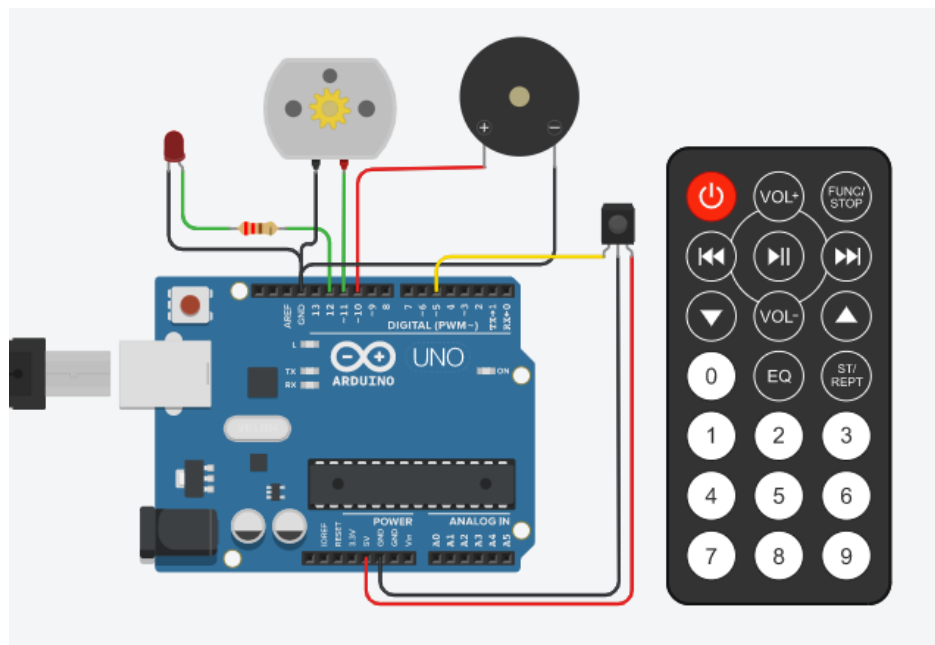
```

// quay nguoc chieu kim dong ho
digitalWrite(8, LOW);
digitalWrite(7, HIGH);
// toc do
analogWrite(11, 150);
delay(5000);
}

```

Câu 10: Điều khiển motor và loa bằng hệ thống hồng ngoại

Sơ đồ mạch điện



Hình 10: Sơ đồ đấu nối mạch điện điều khiển motor và loa bằng hệ thống hồng ngoại

Mô tả sơ đồ hoạt động mạch

- Nhấn vol-: loa kêu.
- Nhấn power: led sáng.
- Nhấn số 1: động cơ quay.

Linh kiện mạch

- Arduino Uno R3.
- IR sensor.
- IR remote.
- DC Motor.
- Loa.

Mô tả chi tiết linh kiện mạch

Code

```
#include <IRremote.h>

IRrecv matHongNgoai(5);
decode_results maNhan;

void setup()
{
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(5, INPUT);
  matHongNgoai.enableIRIn(); //Cho phép hoạt động ngay từ đầu
  Serial.begin(9600); // Cấu hình cổng Serial để in mã nhận
}

void loop()
{
  matHongNgoai.decode(&maNhan); //Tiến hành giải mã

  Serial.println(maNhan.value); //in mã để xem
  //Các xử lý với mã code ở đây
  //Power code: 16580863
```

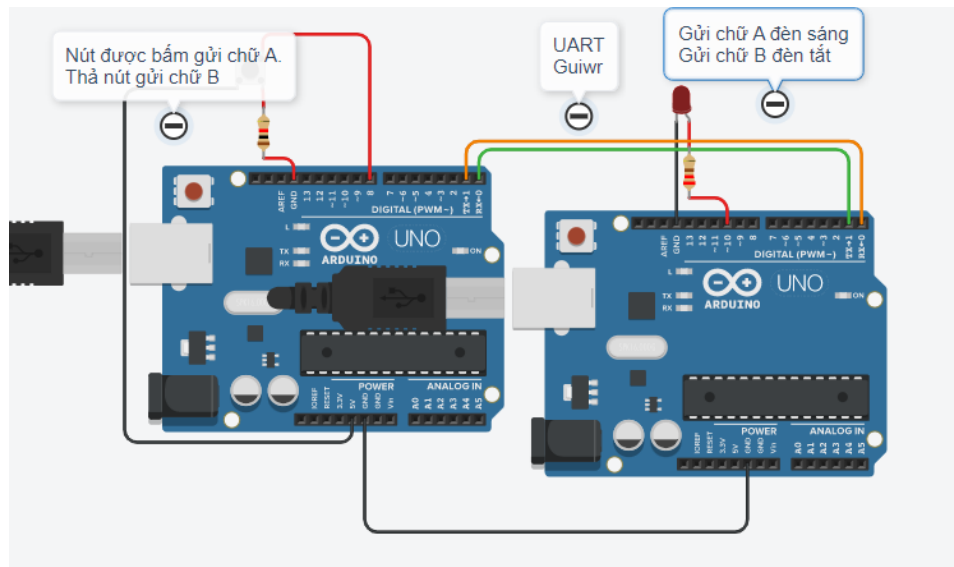
```

//Vol-: 16617583
//so 1: 16582903
if (maNhan.value == 16580863) //phím Power
    digitalWrite(12, HIGH);
if (maNhan.value == 16617583) //phím Vol-
    analogWrite(10, 255); //Kêu to nhất (255) --> nhỏ nhất (0)
if (maNhan.value == 16582903) //phím 1
    analogWrite(11, 255); //Quay nhanh nhất (255) --> chậm nhất (0)
matHongNgoai.resume(); //Tiếp tục lắng nghe
delay(1000);
}

```

Câu 11: Điều khiển đèn led thông qua giao tiếp UART

Sơ đồ mạch điện



Hình 11: Sơ đồ đấu nối mạch điều khiển led qua UART

Mô tả sơ đồ hoạt động mạch

- Nút được bấm gửi chữ A. thì đèn sáng.
- Nút nhả gửi chữ B thì đèn tắt.

Linh kiện mạch

- Arduino Uno R3.
- Pushbutton.
- Resistor.
- Led.

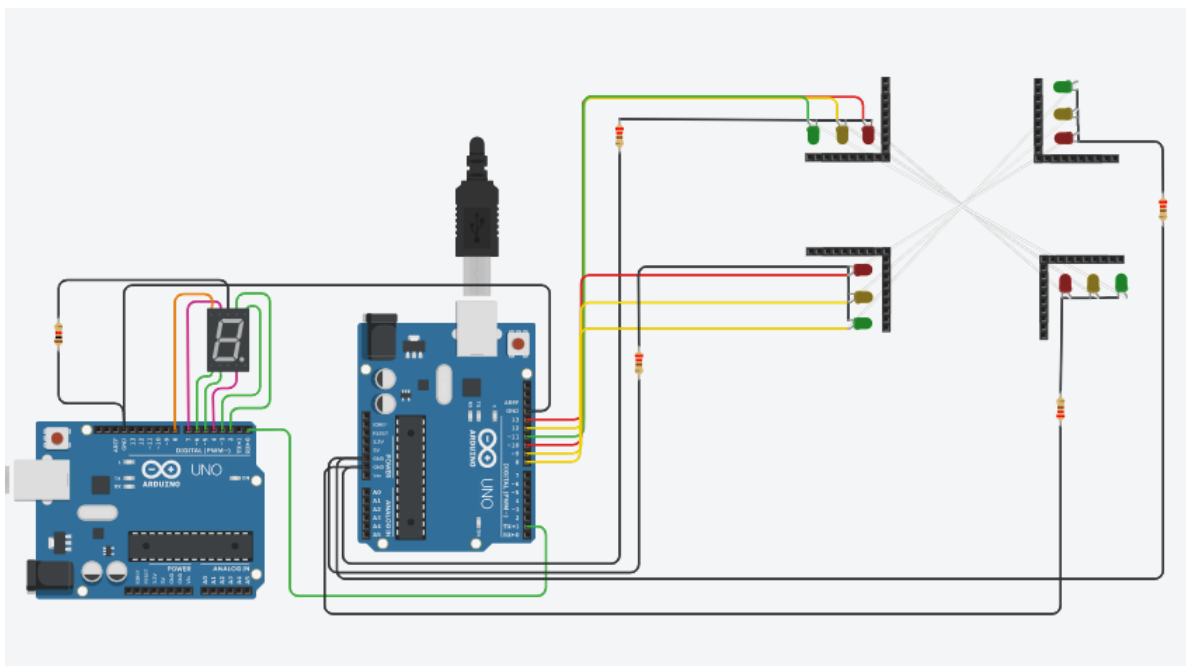
Mô tả chi tiết linh kiện mạch

Code

```
void setup()
{
  pinMode(13,OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  char x = Serial.read();
  if(x=='A')
    digitalWrite(10,HIGH);
  if(x=='B')
    digitalWrite(10,LOW);
  delay(100);
}
```

Câu 12: Led giao thông

Sơ đồ mạch điện



Hình 12: Sơ đồ đấu nối mạch led giao thông

Mô tả sơ đồ hoạt động mạch

- Cứ mỗi 8 giây là đèn sáng đổi từ màu đỏ qua vàng và qua xanh.

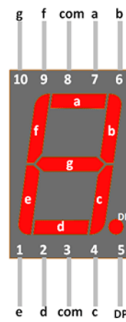
Linh kiện mạch

- Arduino Uno R3.
- Resistor.
- Led.
- Segment Display

Mô tả chi tiết linh kiện mạch

Segment Display

Hiển thị 7 thanh hay **hiển thị 7 đoạn** là thiết bị hiển thị chữ số và ký hiệu khác, có thể là một số chữ Latin, trong các thiết bị điện tử bằng các modul *chữ số hiển thị* (digit) có 7 thanh (hay 7 đoạn), có thể kèm thêm dấu chấm (“.”). Các thanh này được điều khiển mức độ hiện ánh sáng để tạo hình ra chữ số cần hiện .



Code

```
int num_array[10][7] = { { 1,1,1,1,1,1,0 },
                          { 0,1,1,0,0,0,0 },
                          { 1,1,0,1,1,0,1 },
                          { 1,1,1,1,0,0,1 },
                          { 0,1,1,0,0,1,1 },
                          { 1,0,1,1,0,1,1 },
                          { 1,0,1,1,1,1,1 },
                          { 1,1,1,0,0,0,0 },
                          { 1,1,1,1,1,1,1 },
                          { 1,1,1,0,0,1,1 } };
```

```
//function header
```

```
void Num_Write(int);
```

```
void setup()
```

```
{
```

```
  pinMode(0, INPUT);
```

```
// set pin modes
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);

}

void loop()
{

    if( digitalRead(0) == HIGH ) {

        for (int counter = 10; counter > 0; --counter)
        {
            delay(1000);
            Num_Write(counter-1);
        }

    }
    //delay(500);
}
```

```
}
```

```
void Num_Write(int number)
```

```
{
```

```
    int pin= 2;
```

```
    for (int j=0; j < 7; j++) {
```

```
        digitalWrite(pin, num_array[number][j]);
```

```
        pin++;
```

```
    }
```

```
}
```