**MỤC LỤC**

[**Bài 26:** Cho thiết bị Arduino UNO và 1 biến trở, một động cơ Servo được kết nối với thiết bị Arduino, trong đó chân số 2 của biến trở được nối vào một chân analog A0. Hãy thiết kế mạch và viết chương trình lập trình điều khiển cho thiết bị Arduino theo yêu cầu sau: Arduino sẽ đọc giá trị điện áp tại chân analog A0 khi vặn biến trở và chuyển đổi thành góc quay của động cơ Servo theo công thức sau: Góc quay = map(Điện áp, 0,1023,0,180); Thực hiện điều khiển góc quay của động cơ Servo tương ứng với giá trị điện áp chân analog khi vặn biến trở. Hiển thị lên cửa sổ Serial monitor giá trị điện áp và góc quay của động cơ Servo tương ứng. **3**](#_Toc195913972)

[**Bài 29:** Cho thiết bị Arduino UNO, một động cơ Servo và một quang trở được kết nối với thiết bị Arduino. Hãy thiết kế mạch và viết chương trình lập trình điều khiển cho thiết bị Arduino theo yêu cầu sau: Thực hiện điều khiển động cơ servo đóng/mở rèm cửa dựa trên ánh sáng của môi trường chiếu vào quang trở: Arduino sẽ đọc giá trị điện áp tại chân analog của quang trở Khi trời sáng (Giá trị điện áp đọc được >=700) thì động cơ Servo quay về 0 độ để đóng rèm cửa sổ. Khi trời tối (Giá trị điện áp đọc được <700) thì động cơ Servo quay 180 độ để mở rèm cửa sổ. **4**](#_Toc195913973)

[**Bài 30:** Cho thiết bị Arduino UNO, màn hình LCD 16 x 2 kết hợp biến trở, một quang trở và 1 còi báo (buzzer). Hãy thiết kế mạch và viết chương trình lập trình điều khiển cho thiết bị Arduino theo yêu cầu sau: Đọc giá trị ánh sáng từ quang trở, sau đó hiển thị giá trị ánh sáng hiện tại lên màn hình LCD. Nếu giá trị ánh sáng dưới ngưỡng 400, hệ thống sẽ hiển thị cảnh báo: " TOI QUA!" trên màn hình LCD và Bật còi báo (buzzer) để cảnh báo. Nếu ánh sáng trên ngưỡng 400 hiển thị "ANH SANG TOT" trên màn hình LCD và tắt còi báo. **5**](#_Toc195913974)

[**Bài 31:** Cho thiết bị Arduino UNO, một LED 7 đoạn và một nút bấm. Hãy thiết kế mạch và viết chương trình lập trình điều khiển cho thiết bị Arduino theo yêu cầu sau: Khi nhấn nút nhấn, số hiển thị trên LED 7 đoạn tăng lên 1 (0 →1→2→ → 9), khi không nhấn nút nhấn số trên LED 7 đoạn không tăng nữa. Khi tăng đến số 9, nếu nhấn nút tiếp sẽ quay lại 0. **6**](#_Toc195913975)

[**Bài 32:** Cho thiết bị Arduino UNO, màn hình LCD 16 x 2 kết hợp biến trở, một động cơ Servo. Hãy thiết kế mạch và viết chương trình lập trình điều khiển cho thiết bị Arduino theo yêu cầu sau: Ban đầu màn hình LCD hiển thị dòng chữ: “HAY NHAP MK", người dùng nhập mật khẩu vào cửa sổ Serial monitor. Khi người dùng nhập đúng mật khẩu là “123456” màn hình LCD sẽ hiển thị mật khẩu vừa nhập và dòng chữ "MAT KHAU DUNG" và động cơ Servo sẽ quay 180° để mở cửa. Ngược lại khi nhập sai mật khẩu màn hình LCD sẽ hiển thị lại mật khẩu vừa nhập và dòng chữ "SAI MAT KHAU" và động cơ xoay về 0 độ **7**](#_Toc195913976)

[**Bài 38:** Cho thiết bị Arduino UNO. Cho động cơ Servo có chân điều khiển nối vào chân 5. Cho còi Buzzer có chân điều khiển nối vào chân 6. Trong chu kỳ 5 giây, thiết bị sinh một số ngẫu nhiên a từ 0 đến 180. Động cơ sẽ trỏ tới vị trí góc a. Còi sẽ kêu với giá trị a x 100. **9**](#_Toc195913977)

[**Bài 39:** Cho thiết bị Arduino UNO. Cho một nút bấm nối vào chân 5. Cho một thiết bị LED 7 đoạn. Thí sinh tự thiết kế kết nối các chân của LED 7 đoạn với các chân của thiết bị (trừ chân số 5 nối vào nút bấm) để đảm bảo LED 7 đoạn hoạt động được. Khối LED sẽ hiển thị số a được khởi tạo là 0. Khi người dùng bấm nút và nhà ra thì a sẽ tăng lên 1 nếu giá trị a hiện tại lớn hơn 9 hoặc a sẽ bằng 0 nếu giá trị a hiện tại bằng 9. Khối LED 7 đoạn sẽ cập nhập giá trị a. **10**](#_Toc195913978)

[**Bài 40:** Cho thiết bị Arduino UNO. Cho đèn LED RGB với chân điều khiển các LED con R, G, B nối lần lượt vào chân 7, 8, 9. Cho ba nút bấm NR, NG, NB có chân điều khiển nối vào chân 10, 11, 12. Khi nút NR ở trạng thái đóng (người dùng giữ) thì đèn LED con R sẽ sáng, ở trạng thái mở thì đèn R sẽ tắt. Khi nút NG ở trạng thái đóng thì đèn LED con G sẽ sáng, ở trạng thái mở thì đèn G sẽ tắt. Khi nút NB ở trạng thái đóng thì đèn LED con B sẽ sáng, ở trạng thái mở thì đèn LED con B sẽ tắt. **11**](#_Toc195913979)

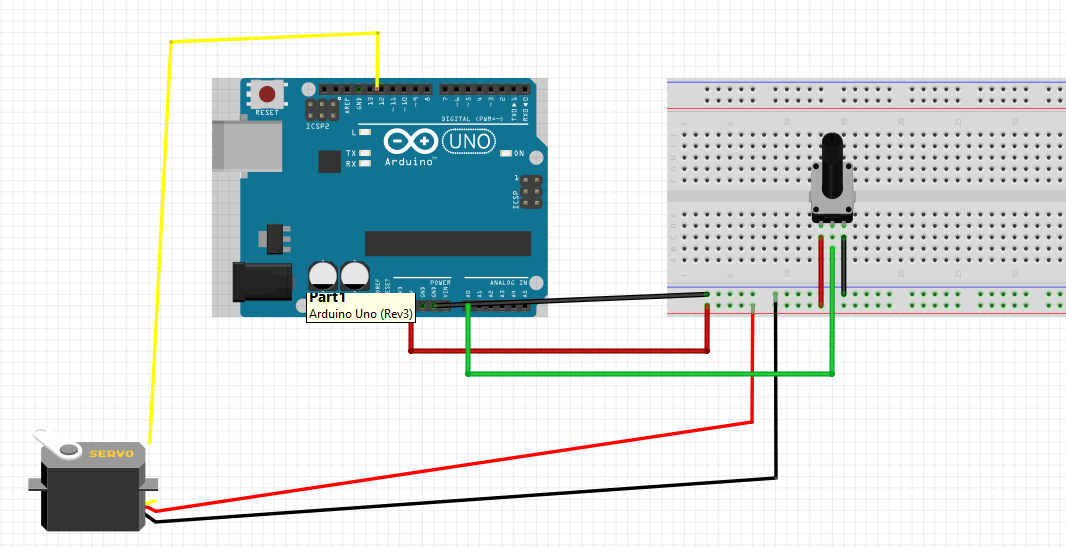
[**Bài 41:** Cho thiết bị Arduino UNO. Cho 3 đèn LED màu đỏ L1, L2, L3 kết nối lần lượt tới các chân điều khiển 3, 4, 5. Khởi tạo các đèn LED đều tắt. Cứ 1 giây thì thiết bị sinh số ngẫu nhiên a từ 1 đến 3. Với a bằng 1 thì đèn L1 sáng, các đèn còn lại đều tất Với a bằng 2 thì đèn L2 sáng, các đèn còn lại đều tắt. Với a bằng 3 thì đèn L3 sáng, các đèn còn lại đều tất. **12**](#_Toc195913980)

[**Bài 42:** Cho thiết bị Arduino UNO. Cho 4 nút bấm B1, B2, B3, B4 kết nối lần lượt tới các chân điều khiển 3, 4, 5, 6. Cho còi Buzzer có chân điều khiển kết nối chân 7. Khi người dùng bấm và nhà một nút thì còi sẽ kêu với giá trị a trong vòng 1 giây. Nếu nút được bấm là B1 thì a = 10, nếu nút được bấm là B2 thì a = 500, nếu nút được bẩm là B3 thì a = 1000, nếu nút bấm là B4 thì a = 6000 **13**](#_Toc195913981)

[**Bài 43:** Cho thiết bị Arduino UNO. Cho một biến trở có chân điều khiển nối vào chân A0, chân điện áp nối vào chân 5V. Cho một động cơ Servo có chân điều khiển nổi vào chân 5. Khi người dùng vận biến trở thì thiết bị sẽ thu nhận giá trị này và gán vào biến a. Sau đó động cơ Servo sẽ gạt đến góc b với công thức b = (a \* 180)/1023 **14**](#_Toc195913982)

[**Bài 44:** Cho thiết bị Arduino UNO. Cho 4 nút bấm B1, B2, B3, B4 kết nối lần lượt tới các chân điều khiển 3, 4, 5, 6. Cho động cơ Servo có chân điều khiển kết nối chân 7. Khi người dùng bấm và nhà một nút thì động cơ sẽ trỏ đến vị trí giá trị a trong vòng 3 giây rồi trở lại vị trí ban đầu. Nếu nút được bấm là B1 thì a = 45, nếu nút được bẩm là B2 thì a = 90, nếu nút được bấm là B3 thì a = 135, nếu nút bấm là B4 thì a = 180 **16**](#_Toc195913983)

# **Bài 26: Cho thiết bị Arduino UNO và 1 biến trở, một động cơ Servo được kết nối với thiết bị Arduino, trong đó chân số 2 của biến trở được nối vào một chân analog A0. Hãy thiết kế mạch và viết chương trình lập trình điều khiển cho thiết bị Arduino theo yêu cầu sau: Arduino sẽ đọc giá trị điện áp tại chân analog A0 khi vặn biến trở và chuyển đổi thành góc quay của động cơ Servo theo công thức sau: Góc quay = map(Điện áp, 0,1023,0,180); Thực hiện điều khiển góc quay của động cơ Servo tương ứng với giá trị điện áp chân analog khi vặn biến trở. Hiển thị lên cửa sổ Serial monitor giá trị điện áp và góc quay của động cơ Servo tương ứng.**



#include <Servo.h>

Servo dtServo;

void setup() {

dtServo.attach(3);

Serial.begin(9600);

}

void loop() {

int giatri = analogRead(A0);

int goc = map(giatri, 0, 1023, 0, 180);

dtServo.write(goc);

float dientap = giatri \* (5.0 / 1023.0);

Serial.print("Dien ap (V): ");

Serial.print(dientap, 2);

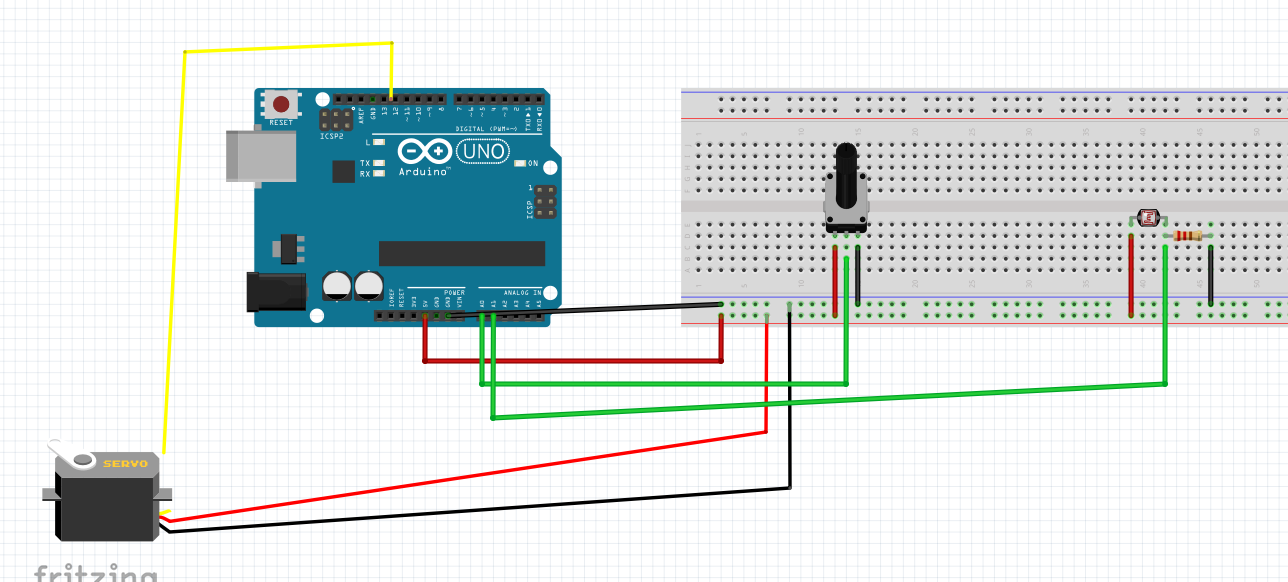
Serial.print(" | Goc servo (do): ");

Serial.println(goc);

delay(500);

}

# **Bài 29: Cho thiết bị Arduino UNO, một động cơ Servo và một quang trở được kết nối với thiết bị Arduino. Hãy thiết kế mạch và viết chương trình lập trình điều khiển cho thiết bị Arduino theo yêu cầu sau: Thực hiện điều khiển động cơ servo đóng/mở rèm cửa dựa trên ánh sáng của môi trường chiếu vào quang trở: Arduino sẽ đọc giá trị điện áp tại chân analog của quang trở Khi trời sáng (Giá trị điện áp đọc được >=700) thì động cơ Servo quay về 0 độ để đóng rèm cửa sổ. Khi trời tối (Giá trị điện áp đọc được <700) thì động cơ Servo quay 180 độ để mở rèm cửa sổ.**



#include <Servo.h>

Servo remCua;

int quangTro = A0;

int servoPin = 9;

void setup() {

Serial.begin(9600);

remCua.attach(servoPin);

Serial.println("Khoi dong he thong dieu khien rem cua bang anh sang");

}

void loop() {

int giaTriSang = analogRead(quangTro);

Serial.print("Gia tri quang tro: ");

Serial.println(giaTriSang);

if (giaTriSang >= 700) {

remCua.write(0);

Serial.println("Troi sang -> dong rem (0°)");

} else {

remCua.write(180);

Serial.println("Troi toi -< mo rem(180°)");

}

delay(1000);

}

# **Bài 30: Cho thiết bị Arduino UNO, màn hình LCD 16 x 2 kết hợp biến trở, một quang trở và 1 còi báo (buzzer). Hãy thiết kế mạch và viết chương trình lập trình điều khiển cho thiết bị Arduino theo yêu cầu sau: Đọc giá trị ánh sáng từ quang trở, sau đó hiển thị giá trị ánh sáng hiện tại lên màn hình LCD. Nếu giá trị ánh sáng dưới ngưỡng 400, hệ thống sẽ hiển thị cảnh báo: " TOI QUA!" trên màn hình LCD và Bật còi báo (buzzer) để cảnh báo. Nếu ánh sáng trên ngưỡng 400 hiển thị "ANH SANG TOT" trên màn hình LCD và tắt còi báo.**

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 4, 5, 6, 7);

int chananalog= A0;

int buzzerPin = 3;

void setup() {

lcd.begin(16, 2);

pinMode(buzzerPin, OUTPUT);

Serial.begin(9600);

}

void loop() {

int giaTriAnhSang = analogRead(chananalog);

Serial.print("Gia tri anh sang: ");

Serial.println(giaTriAnhSang);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Anh sang: ");

lcd.print(giaTriAnhSang);

if (giaTriAnhSang < 400) {

lcd.setCursor(0, 1);

lcd.print("TOI QUA!");

digitalWrite(buzzerPin, HIGH);

} else {

lcd.setCursor(0, 1);

lcd.print("ANH SANG TOT");

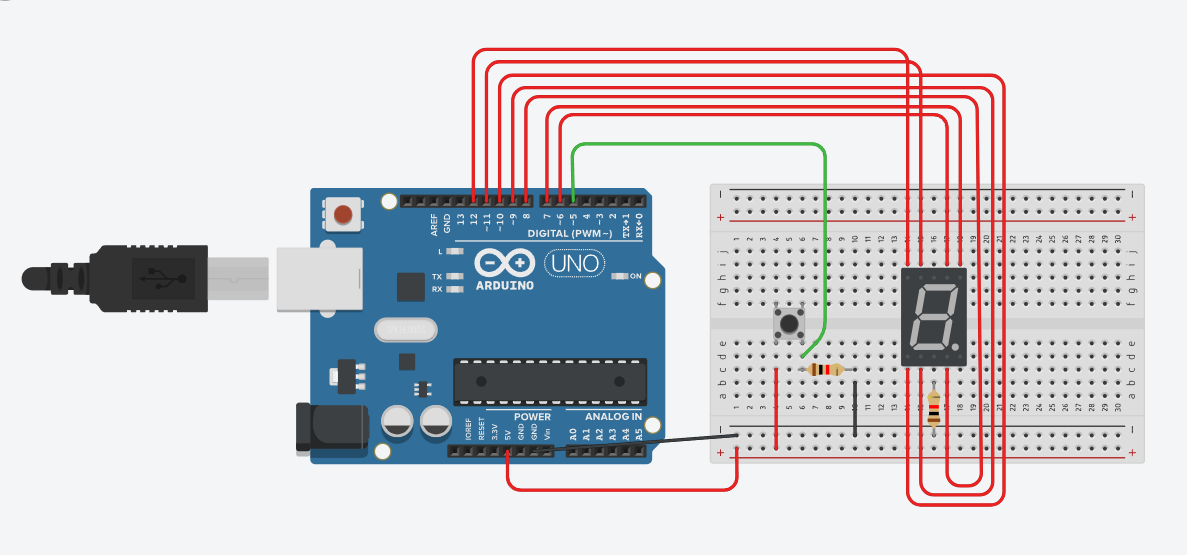
digitalWrite(buzzerPin, LOW);

}

delay(1000);

}

# **Bài 31: Cho thiết bị Arduino UNO, một LED 7 đoạn và một nút bấm. Hãy thiết kế mạch và viết chương trình lập trình điều khiển cho thiết bị Arduino theo yêu cầu sau: Khi nhấn nút nhấn, số hiển thị trên LED 7 đoạn tăng lên 1 (0 →1→2→ → 9), khi không nhấn nút nhấn số trên LED 7 đoạn không tăng nữa. Khi tăng đến số 9, nếu nhấn nút tiếp sẽ quay lại 0.**



int segmentPins[7] = {2, 3, 4, 5, 6, 7, 8}; // a, b, c, d, e, f, g

const byte digitCodes[10][7] = {

{1,1,1,1,1,1,0}, // 0

{0,1,1,0,0,0,0}, // 1

{1,1,0,1,1,0,1}, // 2

{1,1,1,1,0,0,1}, // 3

{0,1,1,0,0,1,1}, // 4

{1,0,1,1,0,1,1}, // 5

{1,0,1,1,1,1,1}, // 6

{1,1,1,0,0,0,0}, // 7

{1,1,1,1,1,1,1}, // 8

{1,1,1,1,0,1,1} // 9

};

int buttonPin = 9;

int currentNumber = 0;

bool lastButtonState = LOW;

void setup() {

for (int i = 0; i < 7; i++) {

pinMode(segmentPins[i], OUTPUT);

}

pinMode(buttonPin, INPUT);

displayDigit(currentNumber);

}

void loop() {

bool buttonState = digitalRead(buttonPin);

if (buttonState == HIGH && lastButtonState == LOW) {

currentNumber = (currentNumber + 1) % 10;

displayDigit(currentNumber);

delay(200);

}

lastButtonState = buttonState;

}

void displayDigit(int num) {

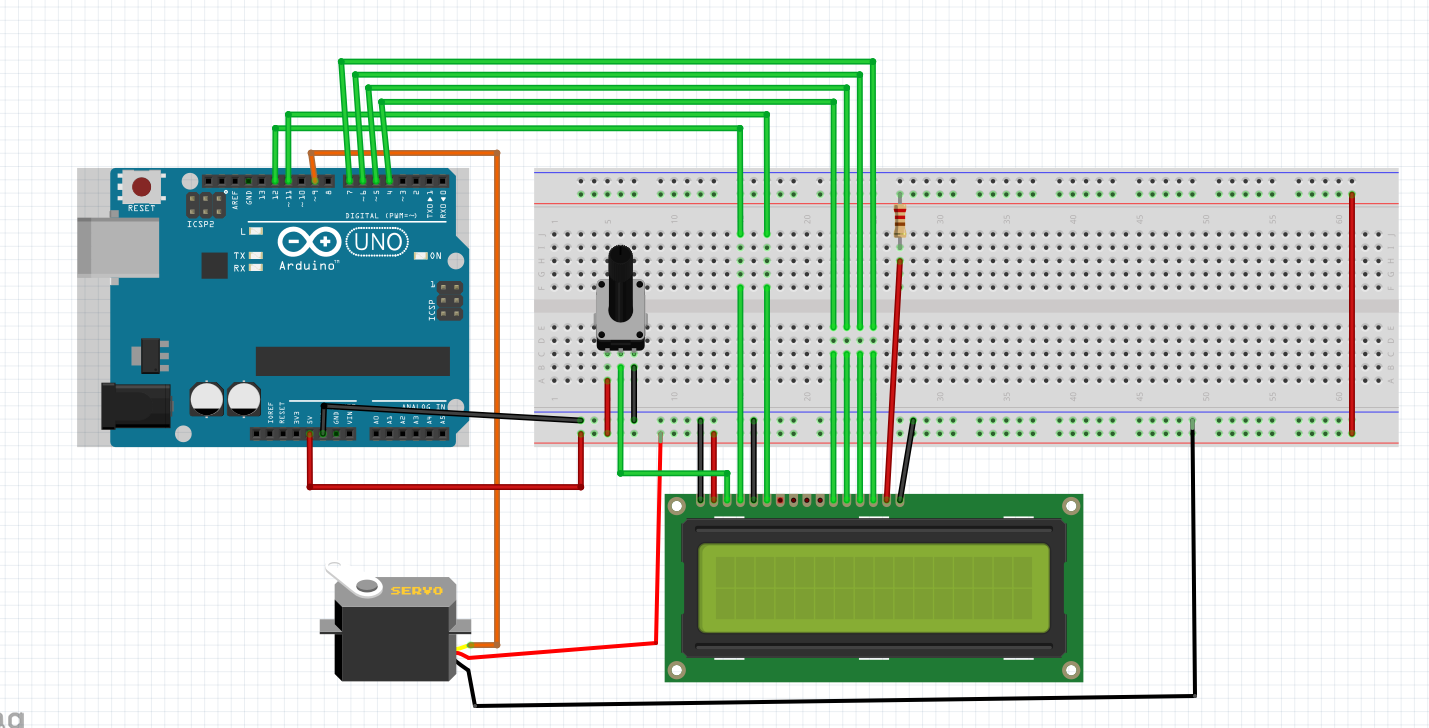
for (int i = 0; i < 7; i++) {

digitalWrite(segmentPins[i], digitCodes[num][i]);

}

}

# **Bài 32: Cho thiết bị Arduino UNO, màn hình LCD 16 x 2 kết hợp biến trở, một động cơ Servo. Hãy thiết kế mạch và viết chương trình lập trình điều khiển cho thiết bị Arduino theo yêu cầu sau: Ban đầu màn hình LCD hiển thị dòng chữ: “HAY NHAP MK", người dùng nhập mật khẩu vào cửa sổ Serial monitor. Khi người dùng nhập đúng mật khẩu là “123456” màn hình LCD sẽ hiển thị mật khẩu vừa nhập và dòng chữ "MAT KHAU DUNG" và động cơ Servo sẽ quay 180° để mở cửa. Ngược lại khi nhập sai mật khẩu màn hình LCD sẽ hiển thị lại mật khẩu vừa nhập và dòng chữ "SAI MAT KHAU" và động cơ xoay về 0 độ**



#include <LiquidCrystal.h>

#include <Servo.h>

LiquidCrystal lcd (12, 11, 4, 5, 6, 7);

Servo myServo;

String password = "123456";

String userInput = "";

void setup() {

Serial.begin(9600);

lcd.begin(16, 2);

myServo.attach(8);

myServo.write(0);

lcd.setCursor(0, 0);

lcd.print("HAY NHAP MK");

Serial.println("Nhap mat khau: ");

}

void loop() {

if (Serial.available() > 0) {

userInput = Serial.readStringUntil('\n');

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(userInput);

lcd.setCursor(0, 1);

if (userInput == password) {

lcd.print("MAT KHAU DUNG");

myServo.write(180);

} else {

lcd.print("SAI MAT KHAU");

myServo.write(0);

}

delay(3000);

lcd.clear();

lcd.setCursor(0, 0);

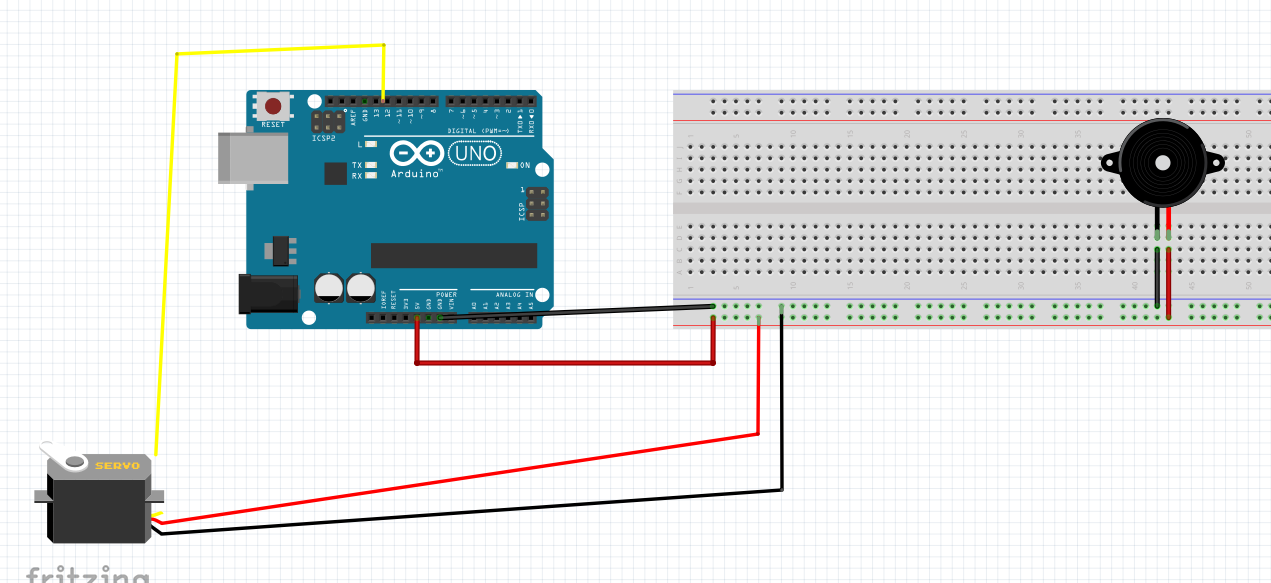
lcd.print("HAY NHAP MK");

Serial.println("Nhap mat khau: ");

}

}

# **Bài 38: Cho thiết bị Arduino UNO. Cho động cơ Servo có chân điều khiển nối vào chân 5. Cho còi Buzzer có chân điều khiển nối vào chân 6. Trong chu kỳ 5 giây, thiết bị sinh một số ngẫu nhiên a từ 0 đến 180. Động cơ sẽ trỏ tới vị trí góc a. Còi sẽ kêu với giá trị a x 100.**



#include <Servo.h>

Servo myServo;

int servoPin = 5;

int buzzerPin = 6;

void setup() {

myServo.attach(servoPin);

randomSeed(analogRead(0));

}

void loop() {

int a = random(0, 181);

myServo.write(a);

int freq = a \* 100;

tone(buzzerPin, freq);

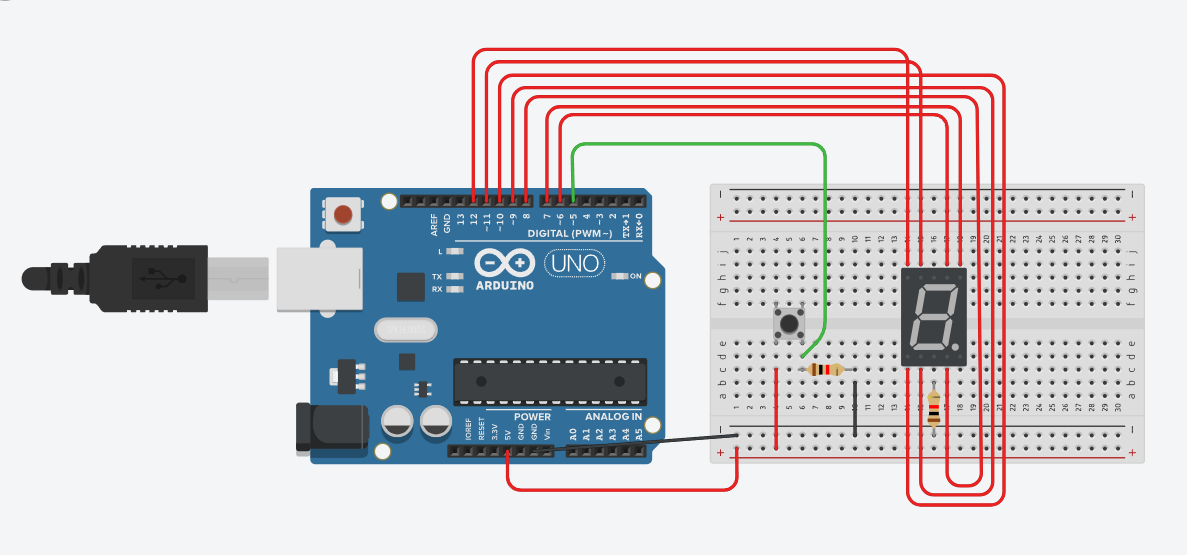
delay(1000);

noTone(buzzerPin);

delay(4000);

}

# **Bài 39: Cho thiết bị Arduino UNO. Cho một nút bấm nối vào chân 5. Cho một thiết bị LED 7 đoạn. Thí sinh tự thiết kế kết nối các chân của LED 7 đoạn với các chân của thiết bị (trừ chân số 5 nối vào nút bấm) để đảm bảo LED 7 đoạn hoạt động được. Khối LED sẽ hiển thị số a được khởi tạo là 0. Khi người dùng bấm nút và nhà ra thì a sẽ tăng lên 1 nếu giá trị a hiện tại lớn hơn 9 hoặc a sẽ bằng 0 nếu giá trị a hiện tại bằng 9. Khối LED 7 đoạn sẽ cập nhập giá trị a.**



int buttonPin = 5;

int segmentPins[7] = {2, 3, 4, 9, 6, 7, 8};

const byte digits[10][7] = {

{1,1,1,1,1,1,0}, // 0

{0,1,1,0,0,0,0}, // 1

{1,1,0,1,1,0,1}, // 2

{1,1,1,1,0,0,1}, // 3

{0,1,1,0,0,1,1}, // 4

{1,0,1,1,0,1,1}, // 5

{1,0,1,1,1,1,1}, // 6

{1,1,1,0,0,0,0}, // 7

{1,1,1,1,1,1,1}, // 8

{1,1,1,1,0,1,1} // 9

};

int a = 0;

bool lastButtonState = LOW;

void setup() {

for (int i = 0; i < 7; i++) {

pinMode(segmentPins[i], OUTPUT);

}

pinMode(buttonPin, INPUT);

displayDigit(a);

}

void loop() {

bool currentButtonState = digitalRead(buttonPin);

if (currentButtonState == HIGH && lastButtonState == LOW) {

delay(50); // debounce

while (digitalRead(buttonPin) == HIGH);

if (a < 9) {

a++;

} else {

a = 0;

}

displayDigit(a);

}

lastButtonState = currentButtonState;

}

void displayDigit(int num) {

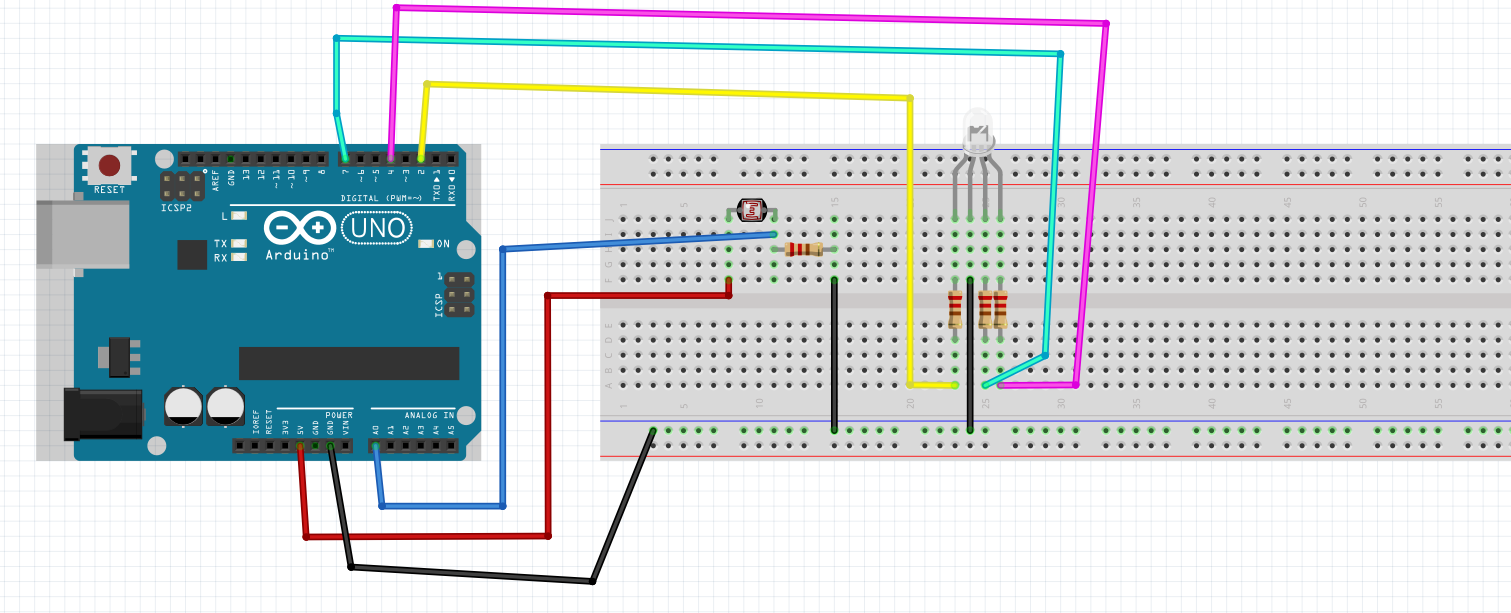
for (int i = 0; i < 7; i++) {

digitalWrite(segmentPins[i], digits[num][i]);

}

}

# **Bài 40: Cho thiết bị Arduino UNO. Cho đèn LED RGB với chân điều khiển các LED con R, G, B nối lần lượt vào chân 7, 8, 9. Cho ba nút bấm NR, NG, NB có chân điều khiển nối vào chân 10, 11, 12. Khi nút NR ở trạng thái đóng (người dùng giữ) thì đèn LED con R sẽ sáng, ở trạng thái mở thì đèn R sẽ tắt. Khi nút NG ở trạng thái đóng thì đèn LED con G sẽ sáng, ở trạng thái mở thì đèn G sẽ tắt. Khi nút NB ở trạng thái đóng thì đèn LED con B sẽ sáng, ở trạng thái mở thì đèn LED con B sẽ tắt.**



// LED RGB pins

int redPin = 7;

int greenPin = 8;

int bluePin = 9;

// Button pins

int NR = 10;

int NG = 11;

int NB = 12;

void setup() {

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

pinMode(bluePin, OUTPUT);

pinMode(NR, INPUT);

pinMode(NG, INPUT);

pinMode(NB, INPUT);

}

void loop() {

bool redState = digitalRead(NR);

bool greenState = digitalRead(NG);

bool blueState = digitalRead(NB);

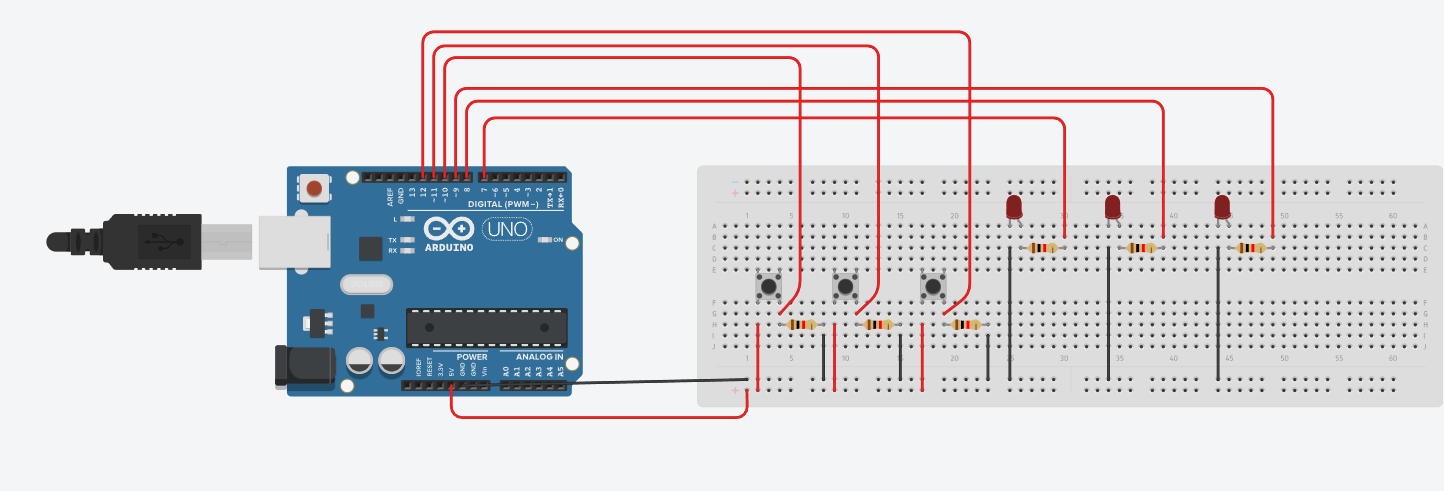
digitalWrite(redPin, redState);

digitalWrite(greenPin, greenState);

digitalWrite(bluePin, blueState);

}

# **Bài 41: Cho thiết bị Arduino UNO. Cho 3 đèn LED màu đỏ L1, L2, L3 kết nối lần lượt tới các chân điều khiển 3, 4, 5. Khởi tạo các đèn LED đều tắt. Cứ 1 giây thì thiết bị sinh số ngẫu nhiên a từ 1 đến 3. Với a bằng 1 thì đèn L1 sáng, các đèn còn lại đều tất Với a bằng 2 thì đèn L2 sáng, các đèn còn lại đều tắt. Với a bằng 3 thì đèn L3 sáng, các đèn còn lại đều tất.**



int L1 = 3;

int L2 = 4;

int L3 = 5;

void setup() {

pinMode(L1, OUTPUT);

pinMode(L2, OUTPUT);

pinMode(L3, OUTPUT);

digitalWrite(L1, LOW);

digitalWrite(L2, LOW);

digitalWrite(L3, LOW);

randomSeed(analogRead(0));

}

void loop() {

int a = random(1, 4);

digitalWrite(L1, LOW);

digitalWrite(L2, LOW);

digitalWrite(L3, LOW);

if (a == 1) {

digitalWrite(L1, HIGH);

} else if (a == 2) {

digitalWrite(L2, HIGH);

} else if (a == 3) {

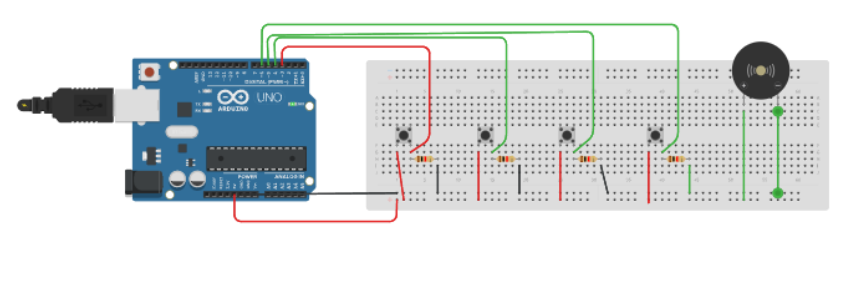
digitalWrite(L3, HIGH);

}

delay(1000);

}

# **Bài 42: Cho thiết bị Arduino UNO. Cho 4 nút bấm B1, B2, B3, B4 kết nối lần lượt tới các chân điều khiển 3, 4, 5, 6. Cho còi Buzzer có chân điều khiển kết nối chân 7. Khi người dùng bấm và nhà một nút thì còi sẽ kêu với giá trị a trong vòng 1 giây. Nếu nút được bấm là B1 thì a = 10, nếu nút được bấm là B2 thì a = 500, nếu nút được bẩm là B3 thì a = 1000, nếu nút bấm là B4 thì a = 6000**



int BTN1 = 3;

int BTN2 = 4;

int BTN3 = 5;

int BTN4 = 6;

int buzzer = 7;

void setup() {

pinMode(BTN1, INPUT);

pinMode(BTN2, INPUT);

pinMode(BTN3, INPUT);

pinMode(BTN4, INPUT);

pinMode(buzzer, OUTPUT);

}

void loop() {

int a = 0;

if (digitalRead(BTN1) == HIGH) {

a = 10;

} else if (digitalRead(BTN2) == HIGH) {

a = 500;

} else if (digitalRead(BTN3) == HIGH) {

a = 1000;

} else if (digitalRead(BTN4) == HIGH) {

a = 6000;

}

if (a > 0) {

tone(buzzer, a);

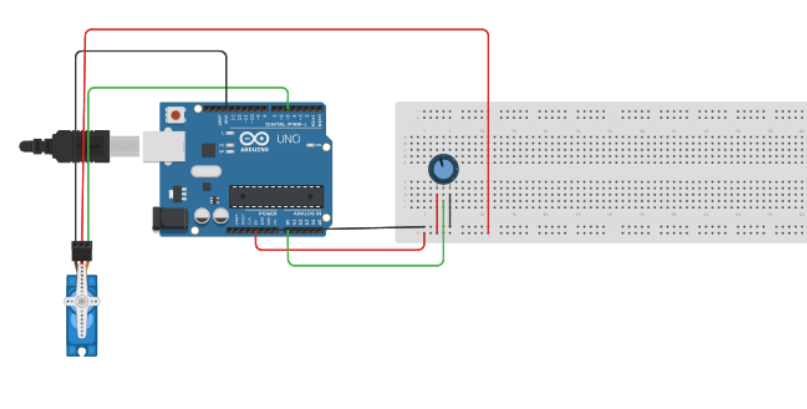
delay(1000);

noTone(buzzer);

}

}

# **Bài 43: Cho thiết bị Arduino UNO. Cho một biến trở có chân điều khiển nối vào chân A0, chân điện áp nối vào chân 5V. Cho một động cơ Servo có chân điều khiển nổi vào chân 5. Khi người dùng vận biến trở thì thiết bị sẽ thu nhận giá trị này và gán vào biến a. Sau đó động cơ Servo sẽ gạt đến góc b với công thức b = (a \* 180)/1023**



#include <Servo.h>

int potPin = A0;

int servoPin = 5;

Servo myServo;

void setup() {

myServo.attach(servoPin);

Serial.begin(9600);

}

void loop() {

int a = analogRead(potPin);

int b = map(a, 0, 1023, 0, 180);

myServo.write(b);

Serial.print("a: ");

Serial.print(a);

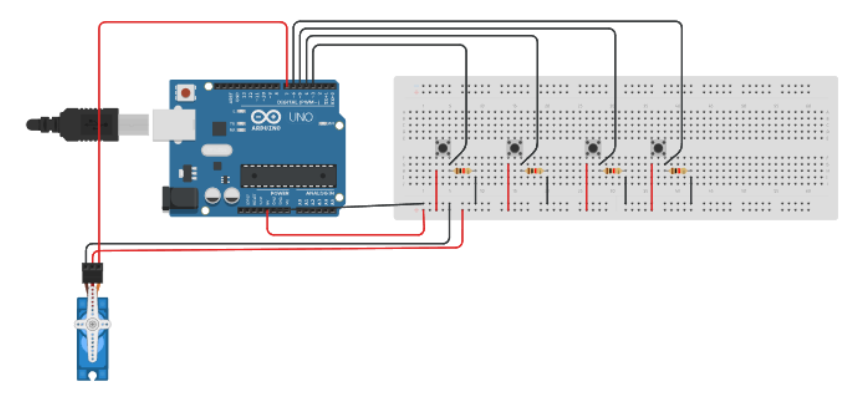
Serial.print(" - b: ");

Serial.println(b);

delay(1500);

}

# **Bài 44: Cho thiết bị Arduino UNO. Cho 4 nút bấm B1, B2, B3, B4 kết nối lần lượt tới các chân điều khiển 3, 4, 5, 6. Cho động cơ Servo có chân điều khiển kết nối chân 7. Khi người dùng bấm và nhà một nút thì động cơ sẽ trỏ đến vị trí giá trị a trong vòng 3 giây rồi trở lại vị trí ban đầu. Nếu nút được bấm là B1 thì a = 45, nếu nút được bẩm là B2 thì a = 90, nếu nút được bấm là B3 thì a = 135, nếu nút bấm là B4 thì a = 180**



#include <Servo.h>

int BTN1 = 3; // Nút B1

int BTN2 = 4; // Nút B2

int BTN3 = 5; // Nút B3

int BTN4 = 6; // Nút B4

int servoPin = 7;

Servo myServo;

void setup() {

pinMode(BTN1, INPUT);

pinMode(BTN2, INPUT);

pinMode(BTN3, INPUT);

pinMode(BTN4, INPUT);

myServo.attach(servoPin);

myServo.write(0);

Serial.begin(9600);

}

void loop() {

if (digitalRead(BTN1) == HIGH) {

moveServo(45);

} else if (digitalRead(BTN2) == HIGH) {

moveServo(90);

} else if (digitalRead(BTN3) == HIGH) {

moveServo(135);

} else if (digitalRead(BTN4) == HIGH) {

moveServo(180);

}

}

void moveServo(int angle) {

myServo.write(angle);

Serial.print("Moving to angle: ");

Serial.println(angle);

delay(3000);

myServo.write(0);

delay(1000);

}