

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙟🕮🙝**

**BÁO CÁO**

**BÀI TẬP LẬP TRÌNH NHÚNG**



|  |  |
| --- | --- |
| **Giáo viên hướng dẫn** | **: Mai Cường Thọ** |
| **Lớp** | **: 61.CNTT-1** |
| **Sinh viên**  **MSSV** | **: Nguyễn Văn Hải Long**  **: 60136035** |

**Nha Trang, năm 2021**

**Phụ Lục**

[**I. Các bài thực hành với Arduino Uno** 2](#_Toc90848718)

[**1. Bật tắt đèn LED đơn giản** 2](#_Toc90848719)

[**2. Nháy đèn LED đơn giản** 3](#_Toc90848720)

[**3. Nút bấm để bật đèn LED** 5](#_Toc90848721)

[**4. Nút bấm để bật đèn LED RBG** 6](#_Toc90848722)

[**5. Điều khiển 2 LED với công tắc khi gạt công tắc thì 1 LED sáng còn 1 LED tối và ngược lại.** 8](#_Toc90848723)

[**6. Điều chỉnh đèn sáng dựa vào cảm biến nhiệt** 10](#_Toc90848724)

[**7. Đọc nhiệt độ ra LCD sử dụng cảm biến nhiệt độ** 12](#_Toc90848725)

[**8. Mạch 10 LED sáng tuần tự** 13](#_Toc90848726)

[**9. Mạch điều khiển LED ma trận 8x8** 15](#_Toc90848727)

[**10. Mạch sử dụng LED 7 đoạn để đếm số từ 0-9** 17](#_Toc90848728)

[**11. Mạch sử dụng LED 7 đoạn để đếm số 0-99** 19](#_Toc90848729)

[**12. Module điều khiển động cơ L298 điều khiển Motor** 20](#_Toc90848730)

[**13. Điều khiển tăng/giảm tốc độ Motor bằng chiết áp** 21](#_Toc90848731)

[**14. Hiển thị tốc độ Motor bằng LCD** 22](#_Toc90848732)

[**15. Máy tính đơn giản sử dụng LCD và Keypad** 24](#_Toc90848733)

[**16. Điều khiển đèn bằng Remote** 27](#_Toc90848734)

[**17. Mô phỏng đèn giao thông** 39](#_Toc90848735)

[**18. Cảm biến khoảng cách điều chỉnh Motor** 41](#_Toc90848736)

[**19. Đo tác dụng lực bằng cảm biến lực** 43](#_Toc90848737)

[**20. Thiết bị bảo mật bằng mật khẩu sử dụng LCD và Keypad** 45](#_Toc90848738)

[**21. Điều chỉnh Micro Servo dựa vào cảm biến Flex** 49](#_Toc90848739)

[**22. Cảm biến PIR điều khiển đèn LED** 51](#_Toc90848740)

[**23. Thiết bị cảnh báo Gas sử dụng cảm biến khí Gas và LED cảnh báo** 52](#_Toc90848741)

[**23. Thiết bị cảnh báo, đo đạt khí Gas sử dụng cảm biến khí Gas và LCD** 54](#_Toc90848742)

[**24. Thiết bị cảnh báo động đất sử dụng cảm biến độ nghiêng** 56](#_Toc90848743)

[**25. Sử dụng quang điện trở để điều khiển đèn LED** 59](#_Toc90848744)

[**26. Sử dụng Board micro:bit điều khiển đèn sáng tuần tự** 60](#_Toc90848745)

[**II. Các bài thực hành với STM32** 62](#_Toc90848746)

[**1. Sử dụng STM32 điều khiển 4 LED** 62](#_Toc90848747)

[**III. Các chương trình IoT** 64](#_Toc90848748)

[**1.Đo nhiệt độ bằng cảm biến sau đó hiển thị trên LCD có sử dụng Arduino** 64](#_Toc90848749)

[**2.Đóng mở cửa với IoT** 65](#_Toc90848750)

[**3.Giám sát nhiệt độ và ra lệnh có điền kiện** 66](#_Toc90848751)

[**4.Bật/tắt thiết bị từ xa thông qua Server** 67](#_Toc90848752)

# **I. Các bài thực hành với Arduino Uno**

## **1. Bật tắt đèn LED đơn giản**

**Mô tả:** Hệ thống được thiết kế để thực hiện bật/tắt đèn LED, Đèn được nối vào Arduino ở cổng số 13, khoản thời gian là 300ms.

**Linh kiện:**

- Arduino Uno

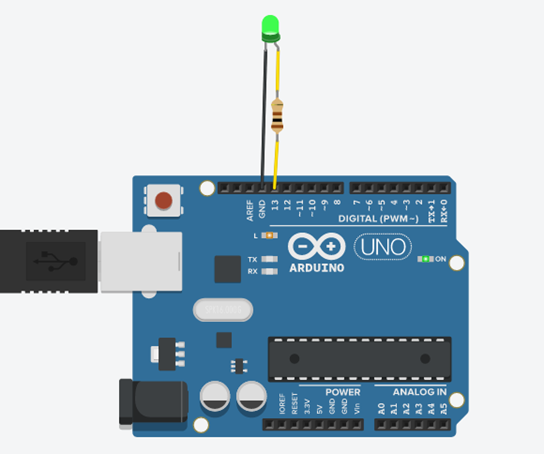
- Điện trở: 100Ω

- Đèn LED

**Code:**

|  |
| --- |
| *void setup()*  *{*  *pinMode(13,OUTPUT);*  *}*  *void loop()*  *{*  *digitalWrite(13,HIGH);*  *delay(300);*  *}* |

**Sơ đồ thiết kế với Tinkercad:**

****

## **2. Nháy đèn LED đơn giản**

**Mô tả:** Hệ thống được thiết kế để thực hiện nháy đèn LED, Đèn được nối vào Arduino ở cổng số 13, khoảng thời gian là 300ms.

**Linh kiện:**

- Arduino Uno

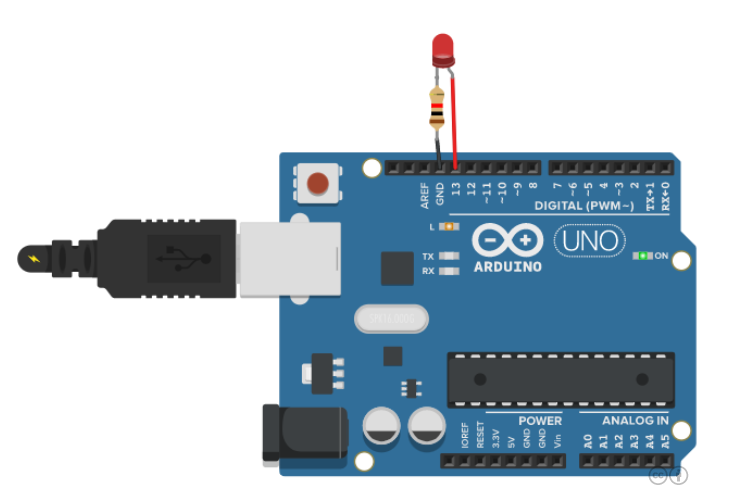
- Điện trở: 100Ω

- Đèn LED

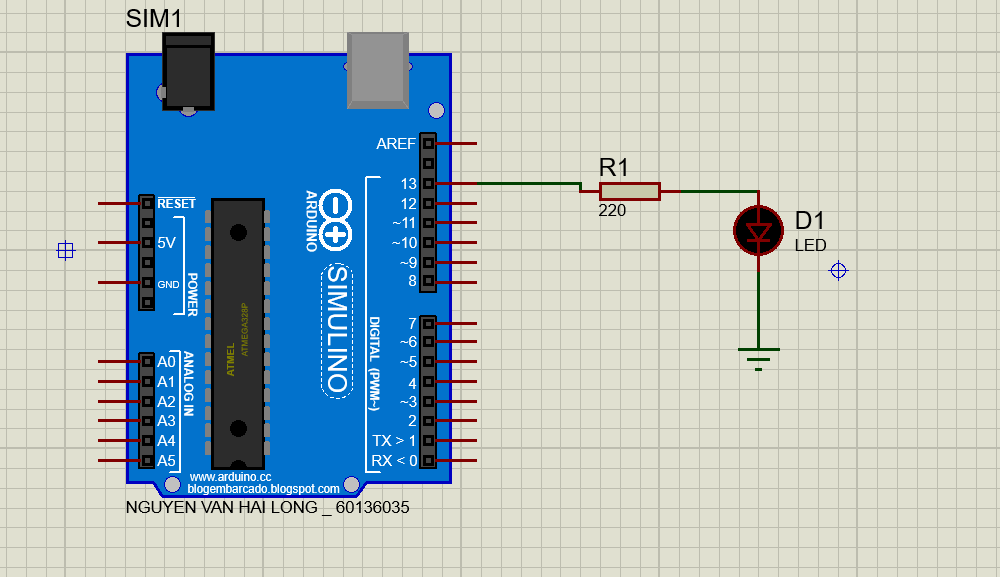
**Code:**

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

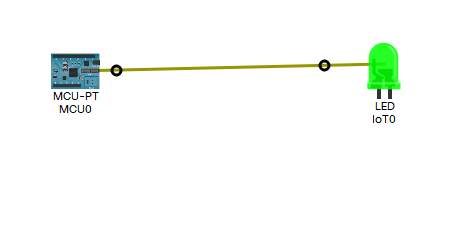
**Sơ đồ thiết kế Tinkercad:**



**Sơ đồ thiết kế Proteus:**



**Sơ đồ thiết kế Cisco Packet Tracer:**



## **3. Nút bấm để bật đèn LED**

**Mô tả:** Hệ thống thực hiện bật LED thông qua nút nhấn, Đèn được kết nối Arduino ở cổng số 12, đầu ra Nút bấm được nối vào pin 2. Đèn sáng khi nhấn nút và ngược lại.

**Linh kiện:**

- Arduino Uno R3

- Điện trở: 100Ω

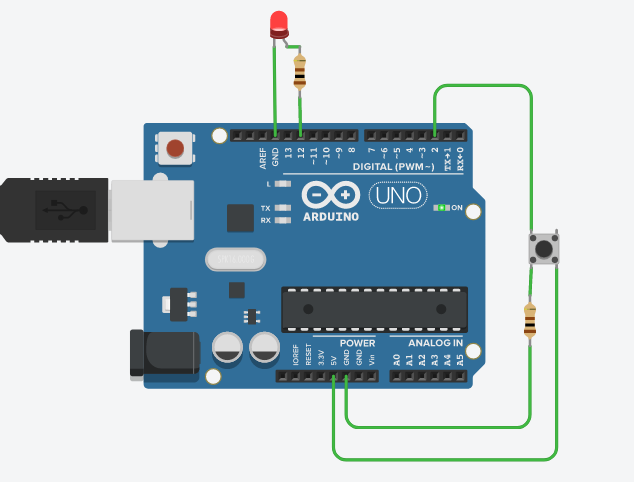
- Đèn LED

- Nút nhấn

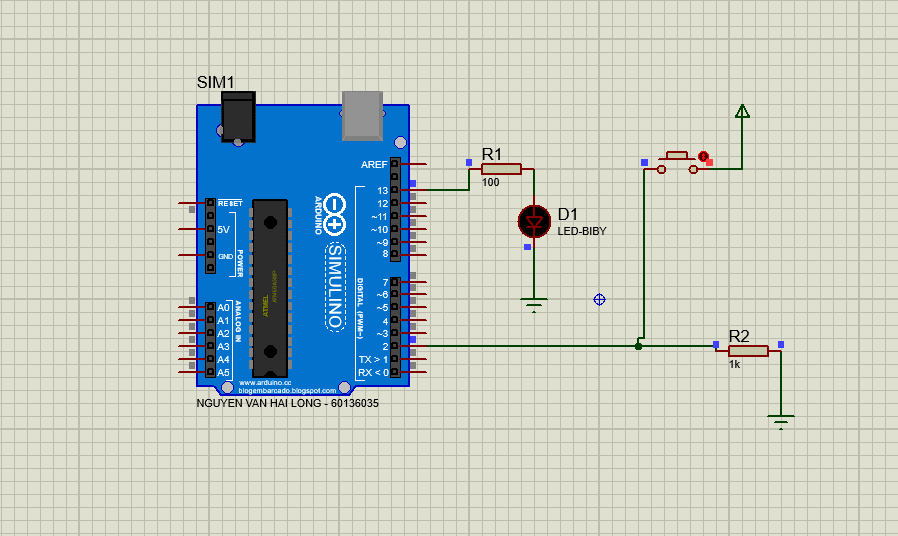
**Code:**

|  |
| --- |
| *double run = false;*  *void setup()*  *{*  *pinMode(12, OUTPUT);*  *pinMode(2, INPUT);*  *Serial.begin(9600);*  *}*  *void loop()*  *{*  *if(digitalRead(2) == HIGH)*  *{*  *delay(400);*  *run = !run;*  *digitalWrite(12,run);*  *Serial.println(run);*  *}*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



**Sơ đồ thiết kế Proteus (đổi thành cổng 13):**



## **4. Nút bấm để bật đèn LED RBG**

**Mô tả:** Hệ thống thực hiện bật LED RBG thông qua nút nhấn, Đèn được kết nối Arduino ở cổng số 12-11-10, đầu ra Nút bấm được nối vào cổng 2. Đèn sáng màu ngẫu nhiên khi nhấn nút và ngược lại.

**Linh kiện:**

- Arduino Uno R3

- Điện trở: 100Ω

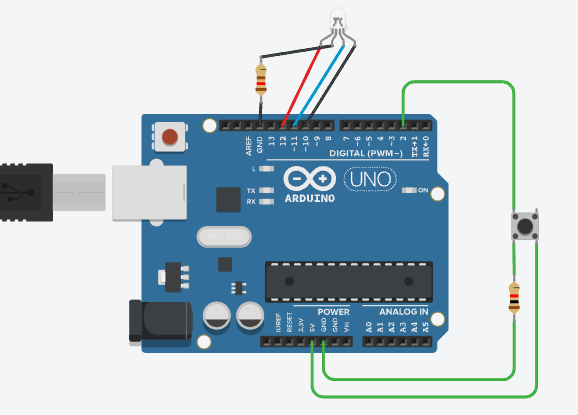
- Đèn LED

- Nút nhấn

**Code:**

|  |
| --- |
| *boolean run = false;*  *void setup()*  *{*  *pinMode(12, OUTPUT);*  *pinMode(11, OUTPUT);*  *pinMode(10, OUTPUT);*  *pinMode(2, INPUT);*  *Serial.begin(9600);*  *}*  *void loop()*  *{*  *if(digitalRead(2) == HIGH)*  *{*  *int res = rand() % (12 - 3 + 1) + 10;/ xuất màu random*  *run = !run;*  *digitalWrite(res,run);*  *Serial.println(run);*  *}*  *else run=0;*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **5. Điều khiển 2 LED với công tắc khi gạt công tắc thì 1 LED sáng còn 1 LED tối và ngược lại.**

**Mô tả:**

Hệ thống thực hiện bật LED thông qua công tắc gạt, đèn 1 được kết nối Arduino ở cổng số 6, đèn 2 được kết nối Arduino ở cổng số 9, đầu ra công tắc bấm được nối vào 1. Một đèn sáng khi nhấn nút (đèn còn lại tắt) và ngược lại**.**

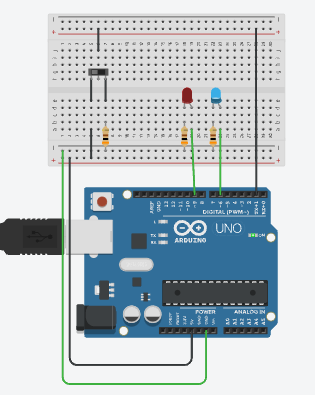
**Linh kiện:**

* Arduino Uno
* Đèn LED: 2
* Điện trở: 3
* Công tắc gạt

**Code:**

|  |
| --- |
| *int Led1=6;*  *int Led2=9;*  *int button=1;*  *void setup() {*  *pinMode(Led1,OUTPUT);*  *pinMode(Led2,OUTPUT);*  *pinMode(button,INPUT);*  *}*  *void loop() {*  *if (digitalRead(button)==1)*  *{digitalWrite(Led1,HIGH);*  *digitalWrite(Led2,LOW);*  *}*  *else*  *{*  *digitalWrite (Led2,HIGH);*  *digitalWrite (Led1,LOW);*    *}*  *}* |

**Sơ đồ thiết kế với Tinkercad :**

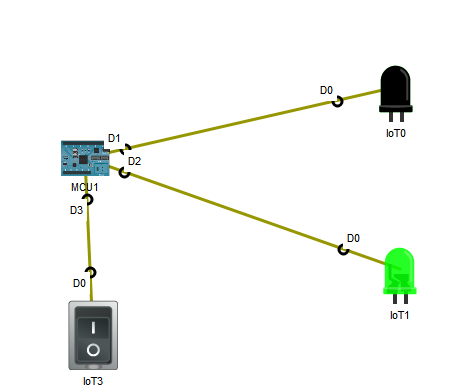


**Cisco Packet Tracer (Công tắc thường – 2 cực):**

**+Code:**

|  |
| --- |
| *function setup() {*  *pinMode(3,INPUT);*  *pinMode(1, OUTPUT);*  *pinMode(2, OUTPUT);*  *Serial.println("Blinking");*  *}*  *function loop() {*  *var x = digitalRead(3);*  *if(x == HIGH)*  *{ digitalWrite(2, LOW);*  *digitalWrite(1, HIGH);*  *}*  *else*  *{*  *digitalWrite(1, LOW);*  *digitalWrite(2, HIGH);*  *}*  *}* |

**+ Sơ đồ thiết kế:**



## **6. Điều chỉnh đèn sáng dựa vào cảm biến nhiệt**

**Mô tả:** Hệ thống nhúng được thiết kế để cho phép điều khiển độ sáng của LED (pin 10) thông qua một cảm biến nhiệt độ (gắn ở A0). Khi nhiệt độ đọc qua cảm biến vượt quá 36 độ C thì đèn sẽ sáng

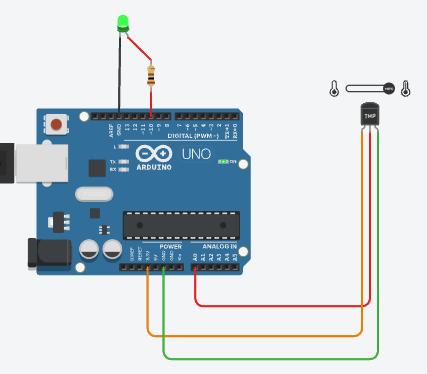
**Linh kiện:**

* Arduino Uno
* LED
* Điện trở
* Cảm biến nhiệt TMP36

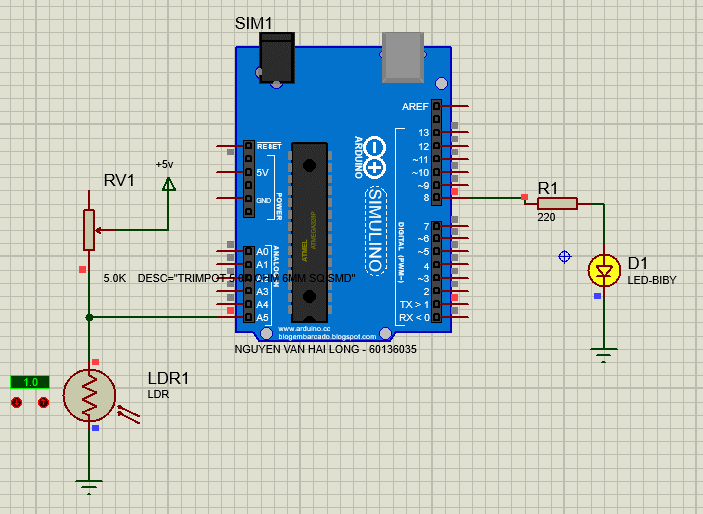
**Code:**

|  |
| --- |
| *void setup()*  *{*  *pinMode(10, OUTPUT);*  *Serial.begin(96000);*  *}*  *void loop()*  *{*  *int x=analogRead(A0);*  *int t= map(x,0,410,-50,150);*    *if (t>=36)*  *digitalWrite(10, HIGH);*  *else*  *digitalWrite(10, LOW);*  *delay(100);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



**Sơ đồ thiết kế Proteus (đổi A0 bằng A5, D10 bằng D8):**



## **7. Đọc nhiệt độ ra LCD sử dụng cảm biến nhiệt độ**

**Mô tả:** Sử dụng LCD để hiển thị nhiệt độ (được đo bằng cảm biến nhiệt)

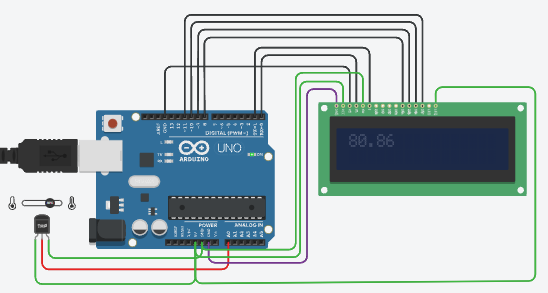
**Linh kiện:**

* Arduino Uno
* Màn hình LCD
* Cảm biến nhiệt độ TMP36
* LCD 16x2

**Code:**

|  |
| --- |
| *#include <LiquidCrystal.h>*  *int sensorPin = 0;*  *LiquidCrystal lcd(0, 1, 8, 9, 10, 11);//Khai báo đầu vào D*  *void setup() {*  *// khai báo thông tin LED*  *lcd.begin(16, 2);*  *}*    *void loop() {*  *int reading = analogRead(sensorPin);*  *float voltage = reading \* 5.0; // nguồn 5V*  *voltage /= 1024.0;*  *//xuất nhiệt độ ra màn hình*  *float temperatureC = (voltage - 0.5) \* 100 ;*  *lcd.print(temperatureC);//in nhiệt độ*  *lcd.setCursor(0, 0);//vị trí in*  *delay(1000);*  *lcd.clear();*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **8. Mạch 10 LED sáng tuần tự**

**Mô tả:** Mạch gồm 10 LED sáng theo thứ tự từ 1 đến 10 (chân tín hiệu từ 13-4)

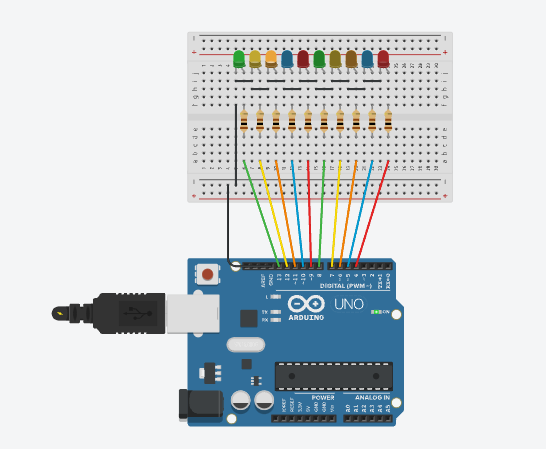
**Linh kiện:**

* Arduino Uno R3
* 10 LED khác màu
* 10 Điện trở

**Code:**

|  |
| --- |
| *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);*  *pinMode(5, OUTPUT);*  *pinMode(4, OUTPUT);*  *}*  *void loop()*  *{*  *digitalWrite(13, HIGH);*  *delay(60);*  *digitalWrite(13, LOW);*  *delay(60);*  *digitalWrite(12, HIGH);*  *delay(60);*  *digitalWrite(12, LOW);*  *delay(60);*  *digitalWrite(11, HIGH);*  *delay(60);*  *digitalWrite(11, LOW);*  *delay(60);*  *digitalWrite(10, HIGH);*  *delay(60);*  *digitalWrite(10, LOW);*  *delay(60);*  *digitalWrite(9, HIGH);*  *delay(60);*  *digitalWrite(9, LOW);*  *delay(60);*  *digitalWrite(8, HIGH);*  *delay(60);*  *digitalWrite(8, LOW);*  *delay(60);*  *digitalWrite(7, HIGH);*  *delay(60);*  *digitalWrite(7, LOW);*  *delay(60);*  *digitalWrite(6, HIGH);*  *delay(60);*  *digitalWrite(6, LOW);*  *delay(60);*  *digitalWrite(5, HIGH);*  *delay(60);*  *digitalWrite(5, LOW);*  *delay(60);*  *digitalWrite(4, HIGH);*  *delay(60);*  *digitalWrite(4, LOW);*  *delay(60);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **9. Mạch điều khiển LED ma trận 8x8**

**Mô tả:** Mô phỏng LED ma trận 8x8 hoặc mạch LED có sẵn, bằng cách bật/tắt đèn để hiện thông điệp.

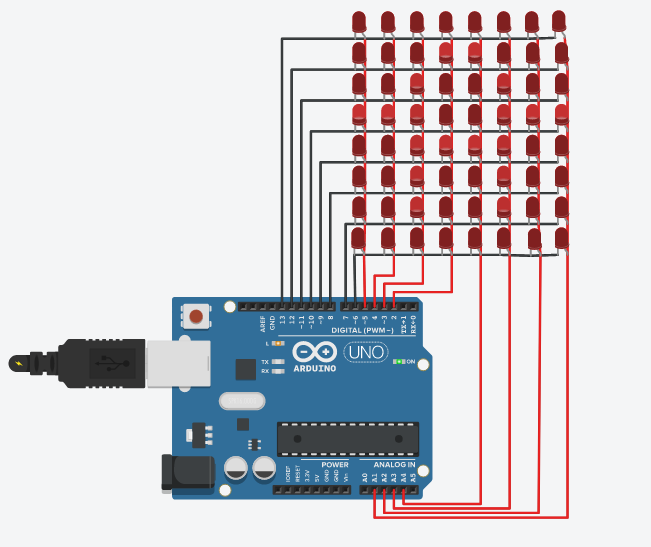
**Linh kiện:**

* Mạch LED 8x8 sẵn hoặc sử dụng 64 LED
* Arduino Uno R3

**Code:**

|  |
| --- |
| *#define ROW1 13*  *#define ROW2 12*  *#define ROW3 11*  *#define ROW4 10*  *#define ROW5 9*  *#define ROW6 8*  *#define ROW7 7*  *#define ROW8 6*  *#define COL1 5*  *#define COL2 4*  *#define COL3 3*  *#define COL4 2*  *#define COL5 A4*  *#define COL6 A3*  *#define COL7 A2*  *#define COL8 A1*  *const int row[] = {ROW1, ROW2, ROW3, ROW4, ROW5, ROW6, ROW7, ROW8};*  *const int col[] = {COL1,COL2, COL3, COL4, COL5, COL6, COL7, COL8};*  *int A[8][8] = {{1,1,1,1,1,1,1,1},*  *{1,1,1,0,0,1,1,1},*  *{1,1,0,1,1,0,1,1},*  *{0,0,0,1,1,0,0,0},*  *{1,1,0,0,0,0,1,1},*  *{1,1,0,1,1,0,1,1},*  *{1,1,0,1,1,0,1,1},*  *{1,1,1,1,1,1,1,1}};*  *void setup() {*  *Serial.begin(9600);*  *for (int i = 2; i <= 13; i++) {*  *pinMode(i, OUTPUT);*  *digitalWrite(i, LOW);*  *}*  *pinMode(A1, OUTPUT);*  *digitalWrite(A1, LOW);*  *pinMode(A2, OUTPUT);*  *digitalWrite(A2, LOW);*  *pinMode(A3, OUTPUT);*  *digitalWrite(A3, LOW);*  *pinMode(A4, OUTPUT);*  *digitalWrite(A4, LOW);*  *}*  *void loop() {*  *delay(0);*  *yaz(A);*  *}*  *void yaz(int matrix[8][8]){*  *for (int c=0; c<8; c++){*  *digitalWrite(col[c], HIGH);*  *for (int r = 0; r < 8; r++){*  *digitalWrite(row[r], 255\*matrix[r][c]);*  *delay(0);*  *}*  *for (int r = 0; r < 8; r++){*  *digitalWrite(row[r], HIGH);*  *delay(0);*  *}*  *digitalWrite(col[c], LOW);*  *}*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **10. Mạch sử dụng LED 7 đoạn để đếm số từ 0-9**

**Mô tả:** Sử dụng Arduino để điều khiển LED 7 đoạn để hiện thị số ở dạng nhị phân, có thể thay đổi theo thời gian. Sử dụng cổng từ vị trí từ 2 đến 9. Sử dụng LED 7 đoạn dạng Cathode.

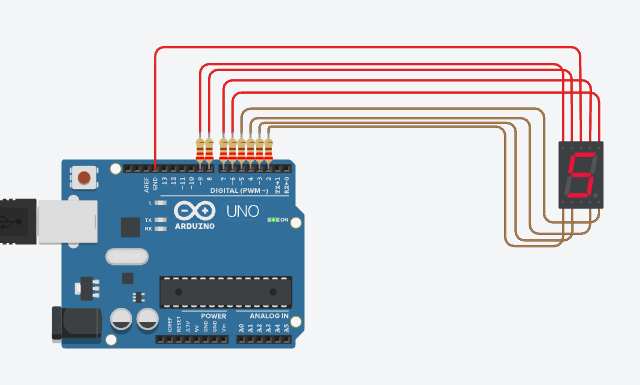
**Linh kiện:**

* Arduino Uno R3
* LED 7 đoạn
* 8 Điện trở: 220Ω

**Code:**

|  |
| --- |
| *byte pins[8] = {2,3,4,6,7,8,9};*  *int duration = 500;*  *byte num0 = 0x3F;*  *byte num1 = 0x0C;*  *byte num2 = 0x5B;*  *byte num3 = 0x5E;*  *byte num4 = 0x6C;*  *byte num5 = 0x76;*  *byte num6 = 0x77;*  *byte num7 = 0x1C;*  *byte num8 = 0x7F;*  *byte num9 = 0x7E;*    *void printN(byte \_digit) {*  *for (int i = 0; i < 8; i++) {*  *digitalWrite(pins[i], bitRead(\_digit, i ));*  *}*  *}*    *void setup() {*  *Serial.begin(9600);*  *for(int i = 0; i <= 9; i++) {*  *pinMode(pins[i], OUTPUT);*  *}*  *}*    *void loop() {*  *printN(num0); delay(1000);*  *printN(num1); delay(1000);*  *printN(num2); delay(1000);*  *printN(num3); delay(1000);*  *printN(num4); delay(1000);*  *printN(num5); delay(1000);*  *printN(num6); delay(1000);*  *printN(num7); delay(1000);*  *printN(num8); delay(1000);*  *printN(num9); delay(1000);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **11. Mạch sử dụng LED 7 đoạn để đếm số 0-99**

**Mô tả:** Sử dụng Arduino để điều khiển 2 LED 7 đoạn để hiện thị số ở dạng nhị phân, có thể thay đổi theo thời gian. Sử dụng cổng từ vị trí 4 đến 11. Sử dụng LED 7 đoạn dạng Cathode. Sử dụng thêm bán dẫn PNP để hạn chế số lượng điện trở.

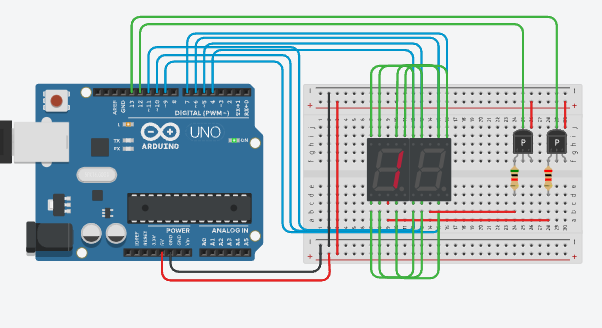
**Linh kiện:**

* Arduino Uno R3
* 2 LED 7 đoạn
* 2 Điện trở: 220Ω
* 2 bán dẫn PNP

**Code:**

|  |
| --- |
| *byte pins[8] = {2,3,4,6,7,8,9};*  *int duration = 500;*  *byte num0 = 0x3F;*  *byte num1 = 0x0C;*  *byte num2 = 0x5B;*  *byte num3 = 0x5E;*  *byte num4 = 0x6C;*  *byte num5 = 0x76;*  *byte num6 = 0x77;*  *byte num7 = 0x1C;*  *byte num8 = 0x7F;*  *byte num9 = 0x7E;*    *void printN(byte \_digit) {*  *for (int i = 0; i < 8; i++) {*  *digitalWrite(pins[i], bitRead(\_digit, i ));*  *}*  *}*    *void setup() {*  *Serial.begin(9600);*  *for(int i = 0; i <= 9; i++) {*  *pinMode(pins[i], OUTPUT);*  *}*  *}*    *void loop() {*  *printN(num0); delay(1000);*  *printN(num1); delay(1000);*  *printN(num2); delay(1000);*  *printN(num3); delay(1000);*  *printN(num4); delay(1000);*  *printN(num5); delay(1000);*  *printN(num6); delay(1000);*  *printN(num7); delay(1000);*  *printN(num8); delay(1000);*  *printN(num9); delay(1000);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **12. Module điều khiển động cơ L298 điều khiển Motor**

**Mô tả:** Module điều khiển động cơ L298 điều khiển Motor qua các cổng 7,8,9

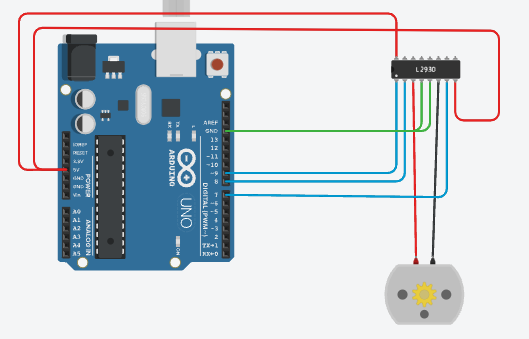
**Linh kiện:**

* Arduino Uno R3
* L298
* Motor

**Code:**

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

**Sơ đồ thiết kế với Tinkercad :**



## **13. Điều khiển tăng/giảm tốc độ Motor bằng chiết áp**

**Mô tả:** Sử dụng chiết áp để điều chỉnh tốc độ Motor

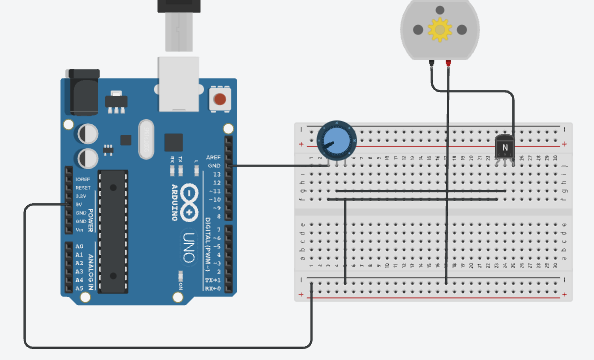
**Linh kiện:**

* Arduino Uno R3
* Chiết áp
* Motor
* NPN Transistor

**Code:**

|  |
| --- |
| *Không code do Arduino chỉ đóng vai trò như nguồn điện* |

**Sơ đồ thiết kế với Tinkercad :**



## **14. Hiển thị tốc độ Motor bằng LCD**

* **Mô tả:** Sử dụng màn hình LCD và các thiết bị khác (Màn đo dao động, Đo điện thế) để thể hiện tốc độ quay của Motor

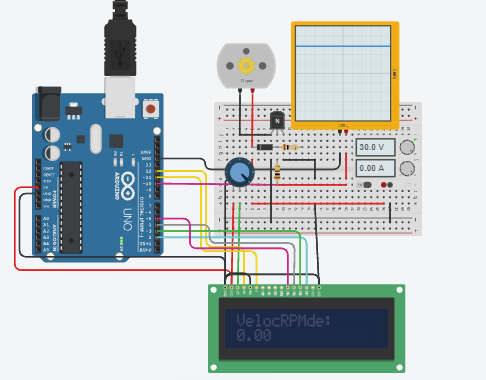
**Linh kiện:**

* Arduino Uno R3
* Điện trở
* Màn đo dao động - 100 ms
* Đo điện thế - 250 kΩ
* Đi-ốt
* Motor
* NPN Transistor
* LCD 16x2

**Code:**

|  |
| --- |
| *#include <LiquidCrystal.h>*  *//static char scrollMessage[] = {"Connect V0->GND "};*  *//int position = 0;*  *volatile int count = 0;*  *int interruptPin = 10;*  *float speedRPM = 0;*  *// khởi tạo thư viện với số lượng các chân giao diện*  *LiquidCrystal lcd(12, 11, 5, 4, 3, 2);*  *// Quy trình dịch vụ ngắt (ISR)*  *void encoderPulse()*  *{*  *count++;*  *}*  *void setup() {*  *lcd.begin(16, 2);*  *lcd.print("Velocidade:");*  *lcd.setCursor(5, 0);*  *lcd.print("RPM");*    *// Thiết lập ngắt bên ngoài*  *pinMode(interruptPin, INPUT\_PULLUP);*  *attachInterrupt(digitalPinToInterrupt(interruptPin), encoderPulse, RISING);*  *}*  *void loop() {*  *lcd.setCursor(0, 1);*  *count = 0;*  *delay(200);*    *// Bộ mã hóa tạo ra ~ 48 xung mỗi vòng quay*  *speedRPM = count \* 6.25;*  *lcd.print(speedRPM);*    *//if (++position > strlen(scrollMessage) - 16) position = 0;*  *// in số giây kể từ khi đặt lại:*  *//lcd.print(&scrollMessage[position]);*  *//delay(100);*  *// điều này là cần thiết để chạy mô phỏng ở tốc độ tối đa*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **15. Máy tính đơn giản sử dụng LCD và Keypad**

**Mô tả:** Sử dụng màn hình LCD và Keypad để mô phỏng một chiếc máy tính bỏ túi. Có thể tính toán 4 phép tính đơn giản

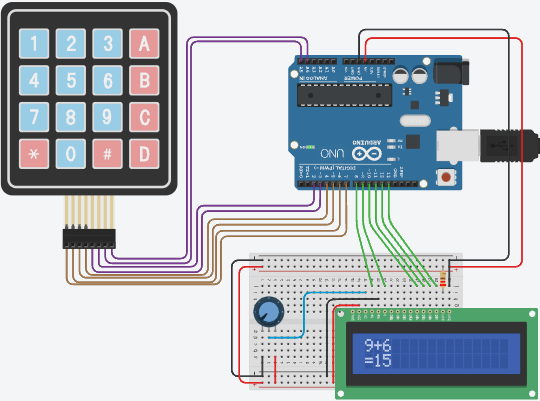
**Linh kiện:**

* LCD 16x2
* Keypad
* Điện trở
* Chiết áp
* Arduino Uno

**Code:**

|  |
| --- |
| *#include <LiquidCrystal.h>*  *#include <Keypad.h>*  *const byte filas = 4; // bốn hàng.*  *const byte columnas = 4; // bốn cột.*  *char tecla[filas][columnas] = {*  *{'1','2','3','A'},*  *{'4','5','6','+'},*  *{'7','8','9','-'},*  *{'\*','0','=','/'}*  *};*  *byte pinFilas[filas] = {7, 6, 5, 4}; // kết nối với các chân ra hàng của bàn phím*  *byte pinColumnas[columnas] = {3, 2, A5, A4}; // kết nối với chân của các cột bàn phím.*  *Keypad keypad = Keypad( makeKeymap(tecla), pinFilas, pinColumnas, filas, columnas );*  *LiquidCrystal lcd(8,9,10,11,12,13); //RS,E,D4,D5,D6,D7*  *String leeCadena1,leeCadena2;*  *byte cuenta=0;*  *char a;*  *long dato1,dato2,rpt1;*  *double dato3,dato4,rpt2;*  *void setup(){*  *lcd.begin(16,2);*  *}*  *void loop(){*  *char key = keypad.getKey();*  *if (key=='0'||key=='1'||key=='2'||key=='3'||key=='4'||key=='5'||key=='6'||key=='7'||key=='8'||key=='9'){*    *if(cuenta==0){*  *lcd.print(key);*  *leeCadena1+=key;*  *dato1=leeCadena1.toInt();*  *}*    *else if(cuenta==1){*  *lcd.print(key);*  *leeCadena2+=key;*  *dato2=leeCadena2.toInt();*  *}*    *}*  *if(key=='A'){*  *leeCadena2="";*  *leeCadena1="";*  *dato1=0;*  *dato2=0;*  *dato3=0;*  *dato4=0;*  *cuenta=0;*  *rpt1=0;*  *rpt2=0;*  *lcd.clear();*  *}*  *else if(key=='='&&a=='+'){*  *cuenta++;*  *lcd.setCursor(0,2);*  *rpt1=dato1+dato2;*  *lcd.print("=");*  *lcd.print(rpt1);*  *}*  *else if(key=='='&&a=='-'){*  *cuenta++;*  *lcd.setCursor(0,2);*  *rpt1=dato1-dato2;*  *lcd.print("=");*  *lcd.print(rpt1);*  *}*  *else if(key=='='&&a=='\*'){*  *cuenta++;*  *lcd.setCursor(0,2);*  *rpt1=dato1\*dato2;*  *lcd.print("=");*  *lcd.print(rpt1);*  *}*  *else if(key=='='&&a=='/'){*  *cuenta++;*  *lcd.setCursor(0,2);*  *dato3=dato1;*  *dato4=dato2;*  *lcd.print("=");*  *rpt2=(dato3/dato4);*  *lcd.print(rpt2);*  *}*    *else if((key=='+'||key=='-'||key=='\*'||key=='/')&&(cuenta==0)){*  *cuenta++;*  *a=key;*  *lcd.print(a);*  *}*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **16. Điều khiển đèn bằng Remote**

**Mô tả:** Sử dụng Remote điều khiển đèn sáng dựa vào IR sensor để nhận tín hiệu. Các phím số khác nhau sẽ điều khiển đèn sáng theo những màu khác nhau.

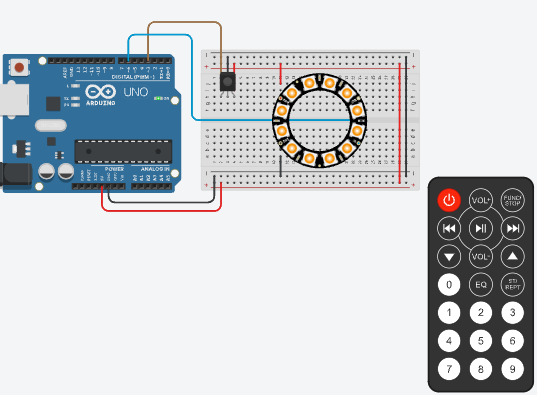
**Linh kiện:**

* Remote
* IR sensor (cảm biến hồng ngoại)
* Đèn chùm dạng vòn 12 LED
* Arduino Uno

**Code:**

|  |
| --- |
| *#include <Adafruit\_NeoPixel.h>*  *#include <IRremote.h>*  *#define PIN 6*  *#define No0 0xFD30CF*  *#define No1 0xFD08F7*  *#define No2 0xFD8877*  *#define No3 0xFD48B7*  *#define No4 0xFD28D7*  *#define No5 0xFDA857*  *#define No6 0xFD6897*  *#define No7 0xFD18E7*  *#define No8 0xFD9867*  *#define No9 0xFD58A7*  *#define Power 0xFD00FF*  *#define Play 0xFDA05F*  *#define EQ 0xFDB04F*  *int RECV\_PIN = 3;*  *IRrecv irrecv(RECV\_PIN);*  *decode\_results results;*  *Adafruit\_NeoPixel strip =*  *Adafruit\_NeoPixel(12, PIN, NEO\_GRB + NEO\_KHZ800);*  *int PixelRainbow[12][3] =*  *{*  *{ 255 , 0 , 0 },{ 255 , 50 , 0 },{ 255 , 255 , 0 },*  *{ 128 , 255 , 0 },{ 0 , 255 , 0 },{ 0 , 100 , 0 },*  *{ 0 , 255 , 255 },{ 0 , 50 , 255 },{ 0 , 0 , 128 },*  *{ 128 , 0 , 255 },{ 128 , 0 , 128 },{ 255 , 0 , 128 }*  *};*  *void setup()*  *{*  *strip.begin();*  *strip.clear();*  *strip.show();*    *Serial.begin(9600);*    *Serial.println("Enabling IRin");*  *irrecv.enableIRIn();*  *Serial.println("Enabled IRin");*  *}*  *void loop()*  *{*      *if (irrecv.decode(&results))*  *{*  *int i=0;*  *if (results.value == Power)*  *{*  *for(i=0 ; i<strip.numPixels() ; i++)*  *{*  *strip.setPixelColor(i,strip.Color(PixelRainbow[i][0],PixelRainbow[i][1],PixelRainbow[i][2]));*  *strip.show();*  *Serial.print("(");*  *Serial.print(PixelRainbow[i][0]);*  *Serial.print(",");*  *Serial.print(PixelRainbow[i][1]);*  *Serial.print(",");*  *Serial.print(PixelRainbow[i][2]);*  *Serial.println(")");*  *irrecv.decode(&results);*  *irrecv.resume();*  *delay(50);*    *if (results.value != Power)*  *goto Idle;*  *}*  *}*  *strip.clear();*      *int j,k;*    *if (results.value == No0)*  *{*  *Serial.println("Running Rainbow");*  *while(results.value == No0)*  *{*  *for(k=255 ; k>0 ; k-=15)*  *{*  *Serial.print("(");*  *Serial.print(k);*  *Serial.print(",");*  *Serial.print(255-k);*  *Serial.print(",");*  *Serial.print(0);*  *Serial.println(")");*  *colorWipe(strip.Color(k,255-k,0), 10);*  *irrecv.decode(&results);*  *irrecv.resume();*  *strip.clear();*    *if (results.value != No0)*  *goto Idle;*  *}*  *for(k=255 ; k>0 ; k-=15)*  *{*  *Serial.print("(");*  *Serial.print(0);*  *Serial.print(",");*  *Serial.print(k);*  *Serial.print(",");*  *Serial.print(255-k);*  *Serial.println(")");*  *colorWipe(strip.Color(0,k,255-k), 10);*  *irrecv.decode(&results);*  *irrecv.resume();*  *strip.clear();*    *if (results.value != No0)*  *goto Idle;*  *}*  *for(k=255 ; k>0 ; k-=15)*  *{*  *Serial.print("(");*  *Serial.print(255-k);*  *Serial.print(",");*  *Serial.print(0);*  *Serial.print(",");*  *Serial.print(k);*  *Serial.println(")");*  *colorWipe(strip.Color(255-k,0,k), 10);*  *irrecv.decode(&results);*  *irrecv.resume();*  *strip.clear();*    *if (results.value != No0)*  *goto Idle;*  *}*  *if (results.value != No0)*  *goto Idle;*  *}*  *}*    *if (results.value == No1)*  *{*  *Serial.println("Running Red");*  *while(results.value == No1)*  *{*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i,strip.Color(255,0,0));*  *strip.show();*  *delay(50);*  *//irrecv.decode(&results);*  *//irrecv.resume();*  *if (irrecv.decode(&results))*  *if (results.value != No1)*  *break;*  *}*      *}*  *goto Idle;*  *}*    *if (results.value == No2)*  *{*  *Serial.println("Running Orange");*  *while(results.value == No2)*  *{*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i,strip.Color(255,50,0));*  *strip.show();*  *delay(50);*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != No2)*  *goto Idle;*  *}*  *irrecv.decode(&results);*  *strip.clear();*  *irrecv.resume();*  *if (results.value != No2)*  *goto Idle;*  *}*  *}*    *if (results.value == No3)*  *{*  *Serial.println("Running Yellow");*  *while(results.value == No3)*  *{*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i,strip.Color(255,255,0));*  *strip.show();*  *delay(50);*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != No3)*  *goto Idle;*  *}*  *irrecv.decode(&results);*  *strip.clear();*  *irrecv.resume();*  *if (results.value != No3)*  *goto Idle;*  *}*  *}*    *if (results.value == No4)*  *{*  *Serial.println("Running Green Yellow");*  *while(results.value == No4)*  *{*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i,strip.Color(128,255,0));*  *strip.show();*  *delay(50);*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != No4)*  *goto Idle;*  *}*  *irrecv.decode(&results);*  *strip.clear();*  *irrecv.resume();*  *if (results.value != No4)*  *goto Idle;*  *}*  *}*      *if (results.value == No5)*  *{*  *Serial.println("Running Green");*  *while(results.value == No5)*  *{*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i,strip.Color(0,255,0));*  *strip.show();*  *delay(50);*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != No5)*  *goto Idle;*  *}*  *irrecv.decode(&results);*  *strip.clear();*  *irrecv.resume();*  *if (results.value != No5)*  *goto Idle;*  *}*  *}*    *if (results.value == No6)*  *{*  *Serial.println("Running Light Blue");*  *while(results.value == No6)*  *{*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i,strip.Color(0,50,255));*  *strip.show();*  *delay(50);*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != No6)*  *goto Idle;*  *}*  *irrecv.decode(&results);*  *strip.clear();*  *irrecv.resume();*  *if (results.value != No6)*  *goto Idle;*  *}*  *}*    *if (results.value == No7)*  *{*  *Serial.println("Running Blue");*  *while(results.value == No7)*  *{*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i,strip.Color(0,0,128));*  *strip.show();*  *delay(50);*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != No7)*  *goto Idle;*  *}*  *irrecv.decode(&results);*  *strip.clear();*  *irrecv.resume();*  *if (results.value != No7)*  *goto Idle;*  *}*  *}*    *if (results.value == No8)*  *{*  *Serial.println("Running Purple");*  *while(results.value == No8)*  *{*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i,strip.Color(128,0,255));*  *strip.show();*  *delay(50);*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != No8)*  *goto Idle;*  *}*  *irrecv.decode(&results);*  *strip.clear();*  *irrecv.resume();*  *if (results.value != No8)*  *goto Idle;*  *}*  *}*    *if (results.value == No9)*  *{*  *Serial.println("Running Pink");*  *while(results.value == No9)*  *{*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i,strip.Color(255,0,128));*  *strip.show();*  *delay(50);*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != No9)*  *goto Idle;*  *}*    *irrecv.decode(&results);*  *strip.clear();*  *irrecv.resume();*  *if (results.value != No9)*  *goto Idle;*  *}*  *}*    *if (results.value == Play)*  *{*  *Serial.println("Running Rainbow");*  *while(results.value == Play)*  *{*  *int i,j;*  *for(j=0; j<256; j+=15)*  *{*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i, Wheel((i+j) & 255));*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != Play)*  *goto Idle;*  *}*  *strip.show();*  *delay(100);*  *}*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != Play)*  *goto Idle;*  *}*  *}*    *if (results.value == EQ)*  *{*  *Serial.println("Running RainbowCycle");*  *while(results.value == EQ)*  *{*  *for(j=0; j<256\*3; j++)*  *{*  *for(i=0; i< strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i, Wheel(((i \* 256 / strip.numPixels()) + j) & 255));*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != EQ)*  *goto Idle;*  *}*  *strip.show();*  *delay(50);*  *}*  *irrecv.decode(&results);*  *irrecv.resume();*  *if (results.value != EQ)*  *goto Idle;*  *}*  *}*    *if (results.value == 0xFFFFFFFF)*  *goto Idle;*    *}*  *Idle:*    *strip.clear();*    *delay(50);*  *}*    *void rainbow(uint8\_t wait) {*  *uint16\_t i, j;*  *for(j=0; j<256; j++) {*  *for(i=0; i<strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i, Wheel((i+j) & 255));*  *}*  *strip.show();*  *delay(50);*  *}*  *}*  *void rainbowCycle(uint8\_t wait)*  *{*  *uint16\_t i, j;*  *for(j=0; j<256\*3; j++)*  *{*  *for(i=0; i< strip.numPixels(); i++)*  *{*  *strip.setPixelColor(i, Wheel(((i \* 256 / strip.numPixels()) + j) & 255));*  *}*  *strip.show();*  *delay(50);*  *}*  *}*  *void theaterChaseRainbow(uint8\_t wait)*  *{*  *for (int j=0; j < 256; j++)*  *{*  *for (int q=0; q < 3; q++)*  *{*  *for (uint16\_t i=0; i < strip.numPixels(); i=i+3)*  *{*  *strip.setPixelColor(i+q, Wheel( (i+j) % 255));*  *}*    *strip.show();*  *delay(50);*  *for (uint16\_t i=0; i < strip.numPixels(); i=i+3)*  *{*  *strip.setPixelColor(i+q, 0);*  *}*  *}*  *}*  *}*  *void colorWipe(uint32\_t c, uint8\_t wait)*  *{*  *for(uint16\_t i=0; i<strip.numPixels(); i++)*  *{ //strip.numPixels()*  *strip.setPixelColor(i, c);*  *strip.show();*  *delay(wait);*  *}*  *}*  *uint32\_t Wheel(byte WheelPos)*  *{*  *WheelPos = 255 - WheelPos;*  *if(WheelPos < 85)*  *{*  *return strip.Color(255 - WheelPos \* 3, 0, WheelPos \* 3);*  *}*  *if(WheelPos < 170)*  *{*  *WheelPos -= 85;*  *return strip.Color(0, WheelPos \* 3, 255 - WheelPos \* 3);*  *}*  *WheelPos -= 170;*  *return strip.Color(WheelPos \* 3, 255 - WheelPos \* 3, 0);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **17. Mô phỏng đèn giao thông**

**Mô tả:** Sử dụng 3 LED đỏ - vàng - xanh và LCD để mô phỏng đèn giao thông. Thực hiện việc đếm số và xuất thông báo ra màn hình LCD.

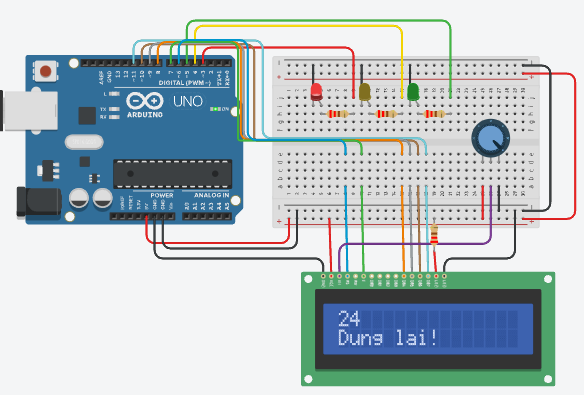
**Linh kiện:**

* 3 LED (Đỏ, Vàng và Xanh Lam)
* Điện trở
* LCD 16x2
* Arduino Uno

**Code:**

|  |
| --- |
| *#include<LiquidCrystal.h>*  *LiquidCrystal MyLCD(6, 7, 8, 9, 10, 11);*  *int red = 3;*  *int yellow = 4;*  *int green = 5;*  *void setup() {*  *pinMode(red, OUTPUT);*  *pinMode(yellow, OUTPUT);*  *pinMode(green, OUTPUT);*  *MyLCD.begin(16, 2);*  *}*  *void yellowBlink(int repeats, int time)*  *{*  *for (int i = repeats; i > 0 ; i--)*  *{*  *MyLCD.clear();*  *MyLCD.setCursor(0, 1);*  *MyLCD.print("Cho doi!");*  *digitalWrite(yellow, HIGH);*  *delay(time);*  *MyLCD.setCursor(0, 0);*  *MyLCD.print(i);*  *digitalWrite(yellow, LOW);*  *delay(time);*  *}*  *}*  *void counter(int count, String Message, int led) {*  *digitalWrite(led, HIGH);*  *for (int i = count ; i > 0; i--) {*  *MyLCD.clear();*  *MyLCD.setCursor(0, 1);*  *MyLCD.print(Message);*  *MyLCD.setCursor(0, 0);*  *MyLCD.print(i);*  *delay(1000);*  *MyLCD.print(" ");*  *if (i < (count - 55)) {*  *MyLCD.print(" ");*  *yellowBlink(count - 55 , 500);*  *break;*  *}*  *}*  *digitalWrite(led, LOW);*  *}*  *void loop() {*  *counter(25, "Dung lai!", red);*  *MyLCD.clear();*  *MyLCD.setCursor(0, 1);*  *MyLCD.print("Doi!");*  *yellowBlink(10, 500);*  *counter(30, "Di chuyen!", green);*  *yellowBlink(10, 500);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **18. Cảm biến khoảng cách điều chỉnh Motor**

**Mô tả:** Sử dụng LCD để hiển thị tốc độ đo được và đưa ra thông số vận tốc theo khoảng cách đo được nhờ đó điều khiển tốc độ motor trong từng trường hợp.

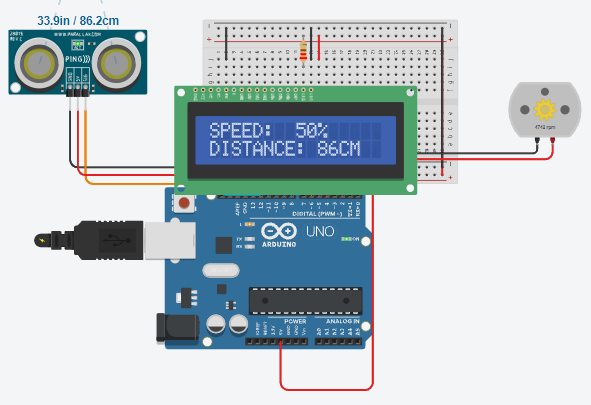
**Linh kiện:**

* LCD 16x2
* Điện trở
* Motor
* Cảm biến khoảng cách
* Arduino Uno

**Code:**

|  |
| --- |
| *int pingIN = 6;*  *int in1Pin = 4;*  *int in2Pin = 2;*  *int enablePin = 5;*  *String msg = " ";*  *int speed = 0;*  *#include <LiquidCrystal.h>*  *LiquidCrystal lcd(13,12,11,10,9,8);*  *void setup() {*  *Serial.begin(9600);*  *pinMode(in1Pin, OUTPUT);*  *pinMode(in2Pin, OUTPUT);*  *pinMode(enablePin, OUTPUT);*  *digitalWrite(in1Pin, HIGH);*  *digitalWrite(in2Pin, LOW);*  *lcd.begin(16,2);*  *lcd.setCursor(0,0);*  *lcd.print("LOADING...");*  *delay(800);*  *lcd.setCursor(0,1);*  *lcd.print("COMPLETE!");*  *delay(1000);*  *lcd.clear();*  *lcd.setCursor(0,0);*  *lcd.print("SPEED: ");*  *lcd.setCursor(0,1);*  *lcd.print("DISTANCE: ");*  *}*  *void loop() {*  *float T, dist, dist\_inted;*  *pinMode(pingIN,OUTPUT);*  *digitalWrite(pingIN, LOW);*  *delayMicroseconds(2);*  *digitalWrite(pingIN, HIGH);*  *delayMicroseconds(5);*  *digitalWrite(pingIN, LOW);*  *pinMode(pingIN, INPUT);*  *T = (pulseIn(pingIN, HIGH))-8;*  *dist = T / 29 / 2 ;*  *dist\_inted = (int) (dist + 0.52);*  *Serial.print("CM: ");*  *Serial.println(dist);*  *delay(100);*  *if(dist\_inted > 100){ speed = 255; msg = "100%";}*  *if(dist\_inted <= 100 && dist\_inted > 50 ){ speed = 127; msg = " 50%";}*  *if(dist\_inted <= 50 && dist\_inted > 25 ){ speed = 63; msg = " 25%";}*  *if(dist\_inted <= 25){ speed = 0; msg = " 0%";}*  *setMotor(speed,msg,dist\_inted);*  *}*  *void setMotor(int speed,String msg,int dist\_inted){*  *analogWrite(enablePin,speed);*  *lcd.setCursor(7,0);*  *lcd.print(msg);*  *lcd.setCursor(10,1);*  *lcd.print(dist\_inted);*  *lcd.print ("CM ");*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **19. Đo tác dụng lực bằng cảm biến lực**

**Mô tả:** Sử dụng màn hình LED để thể hiện các mức độ lực được tác dụng và cảm biến lực để đo lực tác dụng (tương tự trò chơi đấm bốc tính điểm) gồm 6 mức với 6 đèn khác màu. Cảm biến lực sử dụng nguồn 5V và sử dụng chân A0 nhận thông số đo.

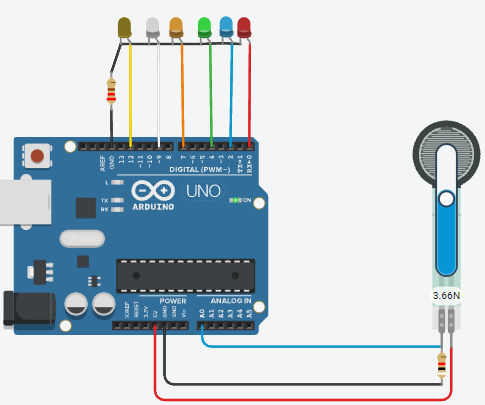
**Linh kiện:**

* Cảm biến lực
* Điện trở
* LED gồm 6 chiếc khác màu
* Arduino Uno

**Code:**

|  |
| --- |
| *#define fsrpin A0*  *#define led1 0*  *#define led2 2*  *#define led3 4*  *#define led4 7*  *#define led5 9*  *#define led6 12*  *int fsrreading;*  *void setup() {*  *Serial.begin(9600);*  *pinMode(led1, OUTPUT);*  *pinMode(led2, OUTPUT);*  *pinMode(led3, OUTPUT);*  *pinMode(led4, OUTPUT);*  *pinMode(led5, OUTPUT);*  *pinMode(led6, OUTPUT);*  *}*  *void loop() {*  *fsrreading = analogRead(fsrpin);*  *Serial.println(fsrreading);*  *if (fsrreading > 200) {*  *digitalWrite(led1, HIGH);*  *}*  *else digitalWrite(led1, LOW);*  *if (fsrreading > 450) {*  *digitalWrite(led2, HIGH);*  *}*  *else digitalWrite(led2, LOW);*  *if (fsrreading > 550) {*  *digitalWrite(led3, HIGH);*  *}*  *else digitalWrite(led3, LOW);*  *if (fsrreading > 650) {*  *digitalWrite(led4, HIGH);*  *}*  *else digitalWrite(led4, LOW);*  *if (fsrreading > 800) {*  *digitalWrite(led5, HIGH);*  *}*  *else digitalWrite(led5, LOW);*  *if (fsrreading > 900) {*  *digitalWrite(led6, HIGH);*  *}*  *else digitalWrite(led6, LOW);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **20. Thiết bị bảo mật bằng mật khẩu sử dụng LCD và Keypad**

**Mô tả:** Sử dụng màn hình LCD và Keypad để mô phỏng thiết bị bảo mật bằng mật mã. Thiết bị cho phép nhập mật mã bảo mật lần đầu (khởi tạo mã) sau đó được nhập nhiều nhất 3 lần (mỗi lần sai thì 1 đèn sẽ tắt). Khi nhập sai quá 3 lần thì in ra báo động.

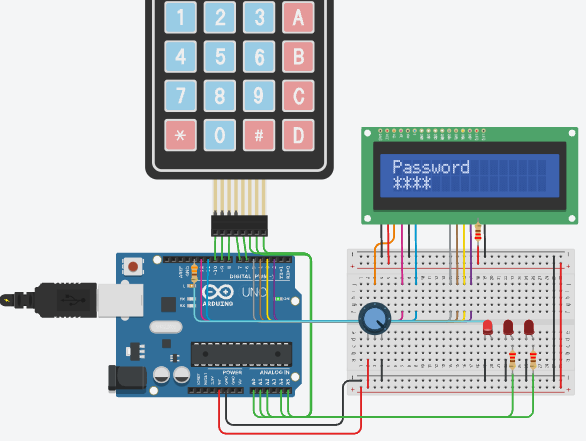
**Linh kiện:**

* LCD 16x2
* Điện trở
* Keypad
* Arduino Uno

**Code:**

|  |
| --- |
| *#include <LiquidCrystal.h>*  *#include<Keypad.h>*  *LiquidCrystal lcd(12, 11, 5, 4, 3, 2);*  *int led[] = {A4, A5, 13};*  *char keys[4][4] = {*  *{'1','2','3','A'},*  *{'4','5','6','B'},*  *{'7','8','9','C'},*  *{'\*','0','#','D'}*  *};*  *byte rowPins[4] = {10,9, 8, 7};*  *byte colPins[4] = {6, A0, A1, A2};*  *Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, 4, 4 );*  *char str[4]={'\*','\*','\*','\*'};*  *char pwd[4]={'\*','\*','\*','\*'};*  *int i=0,j=0,k=0;*  *bool insert=false;*  *void setup() {*  *lcd.begin(16, 2);*  *lcd.print("Insert Password");*  *lcd.setCursor(0, 1);*  *lcd.print("\*\*\*\*");*  *for (int i = 0; i < 3; i++){*  *pinMode(led[i], OUTPUT);*  *}*  *}*  *void loop() {*  *if(insert==false)*  *{*  *char key = keypad.getKey();*  *if (key != NO\_KEY&&i!=4)*  *{*  *str[i]=key;*  *lcd.setCursor(0,1);*  *lcd.print(str[0]);*  *lcd.print(str[1]);*  *lcd.print(str[2]);*  *lcd.print(str[3]);*  *i++;*  *}*  *if(i==4)*  *{*  *lcd.setCursor(6,1);*  *lcd.print("Saving");*  *delay(2000);*  *insert=true;*  *clear();*  *openled();*  *}*  *}*  *else*  *{*  *char key = keypad.getKey();*  *if (key != NO\_KEY&&j!=4)*  *{*  *pwd[j]=key;*  *lcd.setCursor(0,1);*  *lcd.print(pwd[0]);*  *lcd.print(pwd[1]);*  *lcd.print(pwd[2]);*  *lcd.print(pwd[3]);*  *j++;*  *}*  *if(j==4)*  *{*  *lcd.setCursor(6,1);*  *if(pwd[0]==str[0]&&pwd[1]==str[1]&&pwd[2]==str[2]&&pwd[3]==str[3])*  *{*  *lcd.print("Correct!");*  *//delay(2000);*  *//reset();*  *}*  *else*  *{*  *lcd.print("Error!");*  *delay(2000);*  *digitalWrite(led[k],LOW);*  *k++;*  *j=0;*  *if(k==3)*  *{*  *lcd.clear();*  *lcd.print("Alarm!!");*  *delay(500);*  *lcd.display();*  *}*  *else*  *clear();*  *}*  *}*  *}*  *}*  *void clear()*  *{*  *lcd.clear();*  *lcd.setCursor(0,0);*  *lcd.print("Password");*  *lcd.setCursor(0,1);*  *lcd.print("\*\*\*\*");*  *pwd[0]='\*';*  *pwd[1]='\*';*  *pwd[2]='\*';*  *pwd[3]='\*';*  *}*  *void openled()*  *{ for(int i=0;i<3;i++)*  *{*  *digitalWrite(led[i], HIGH);*  *}*  *}*  *void reset()*  *{*  *lcd.clear();*  *i=0,j=0,k=0;*  *insert=false;*  *pwd[0]='\*';*  *pwd[1]='\*';*  *pwd[2]='\*';*  *pwd[3]='\*';*  *str[0]='\*';*  *str[1]='\*';*  *str[2]='\*';*  *str[3]='\*';*  *lcd.print("Insert Password");*  *lcd.setCursor(0, 1);*  *lcd.print("\*\*\*\*");*  *for(int i=0;i<3;i++)*  *{*  *digitalWrite(led[i], LOW);*  *}*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **21. Điều chỉnh Micro Servo dựa vào cảm biến Flex**

**Mô tả:** Điều chỉnh Micro Servo tương ứng với độ cong khi đo được bằng cảm biển Flex. Khi độ cong quá điều kiện cho phép thì Servo sẽ xoay tới góc 180.

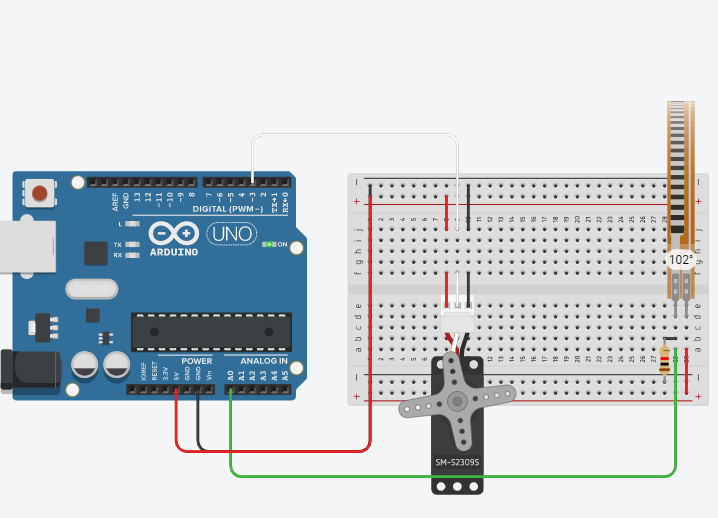
**Linh kiện:**

* Micro Servo (dạng motor góc quay nằm trong khoảng bất kì từ 0o - 180o)
* Điện trở 1k Ω
* Cảm biến Flex (cảm biến cong)
* Arduino Uno

**Code:**

|  |
| --- |
| *#include <Servo.h> // Thư viện Servo Motor*  *#define flex\_sensor A0*  *Servo motorcito;*  *int lectura;*  *void setup()*  *{*  *motorcito.attach(3);*  *Serial.begin(9600);*  *}*  *void loop()*  *{*  *lectura= analogRead (flex\_sensor);*  *Serial.println(lectura);*  *if(lectura>=22)*  *{*  *motorcito.write(0);*  *}*  *else*  *{*  *motorcito.write(180);*  *}*  *delay(300);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **22. Cảm biến PIR điều khiển đèn LED**

**Mô tả:** Cảm biến PIR được nối với cổng 8 và LED cảnh báo sẽ được nối với cổng 10. Cảm biến PIR sẽ phát hiện ra đối tượng phát nhiệt đang di chuyển trong phạm vi từ đó xuất tín hiệu để điều khiển đèn sáng.

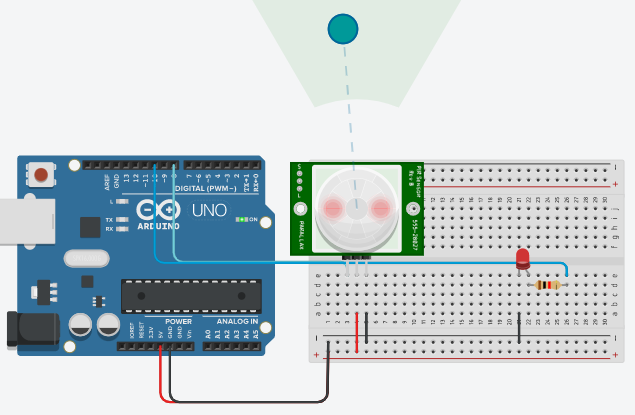
**Linh kiện:**

* Cảm biến PIR (là cảm biến thụ động phát hiện tia hồng ngoại phát ra từ môi trường xung quanh)
* Điện trở
* Đèn LED
* Arduino Uno

**Code:**

|  |
| --- |
| *int LED = 10;*  *int DO\_Sensor = 8;*  *int Sensitivity = 0;*  *int PIR\_state = LOW;*  *void setup() {*  *pinMode(LED, OUTPUT);*  *pinMode(DO\_Sensor, INPUT);*  *Serial.begin(9600);*  *}*  *void loop() {*  *Sensitivity = digitalRead(DO\_Sensor);*  *if (Sensitivity == HIGH) {*  *digitalWrite(LED, HIGH);*  *if (PIR\_state == LOW) {*  *PIR\_state = HIGH;*  *}*  *}else{*  *digitalWrite(LED, LOW);*  *if(PIR\_state == HIGH){*  *PIR\_state = LOW;*  *}*  *}*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **23. Thiết bị cảnh báo Gas sử dụng cảm biến khí Gas và LED cảnh báo**

**Mô tả:** Khi nồng độ chất khí càng cao điện áp đầu ra tại chân A5 của cảm biến càng tăng lên. Tín hiệu tương tự này được trực tiếp xuất ra đầu ra và được đưa tới bộ so sánh LM393 để số hoá tín hiệu. Bộ so sánh cùng với một biến trở để điều chỉnh độ nhạy của cảm biến. Khi nồng độ khí vượt qua giới hạnh điều chỉnh nó sẽ đưa ra tín hiệu dạng số ở cổng 2 đến 4(mỗi đèn sáng là 1 mức độ tương ứng).

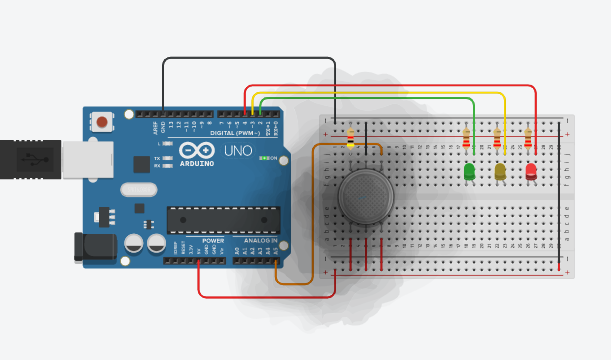
**Linh kiện:**

* Cảm biến khí Gas
* Đèn LED
* Điện trở
* Arduino Uno

**Code:**

|  |
| --- |
| *int led1 = 2;*  *int led2 = 3;*  *int led3 = 4;*  *int sensor = A5;*  *int gas = 0;*  *void setup()*  *{*  *pinMode(led1, OUTPUT);*  *pinMode(led2, OUTPUT);*  *pinMode(led3, OUTPUT);*  *Serial.begin(9600);*  *}*  *void loop()*  *{*  *gas = analogRead(sensor); // Cảm biến được sử dụng là cảm biến MQ4, để đọc Methane*  *Serial.println(gas); // hiển thị kết quả của giá trị gas*  *gas = map(gas, 306, 750, 0, 100); // thay đổi giá trị cảm biến (đọc từ 306 - 740) phạm vi 0-100*    *// điều kiện khi khí ở rất gần cảm biến*  *if (gas >= 60) {*  *digitalWrite(led1, LOW);*  *digitalWrite(led2, LOW);*  *digitalWrite(led3, HIGH);*  *}*  *// điều kiện khi khí tầm trung với cảm biến*  *else if (gas >= 30) {*  *digitalWrite(led1, LOW);*  *digitalWrite(led2, HIGH);*  *digitalWrite(led3, LOW);*  *}*  *// điều kiện khi khí ở xa cảm biến (mức bình thường)*  *else {*  *digitalWrite(led1, HIGH);*  *digitalWrite(led2, LOW);*  *digitalWrite(led3, LOW);*  *}*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **23. Thiết bị cảnh báo, đo đạt khí Gas sử dụng cảm biến khí Gas và LCD**

**Mô tả:** Khi nồng độ chất khí càng cao điện áp đầu ra tại chân A5 của cảm biến càng tăng lên. Tín hiệu tương tự này được trực tiếp xuất ra đầu ra và được đưa tới bộ so sánh LM393 để số hoá tín hiệu. Bộ so sánh cùng với một biến trở để điều chỉnh độ nhạy của cảm biến. Nồng độ khí được ghi lại và xuất ra màn hình LCD với 2 dòng là mức độ và nồng độ đo được.

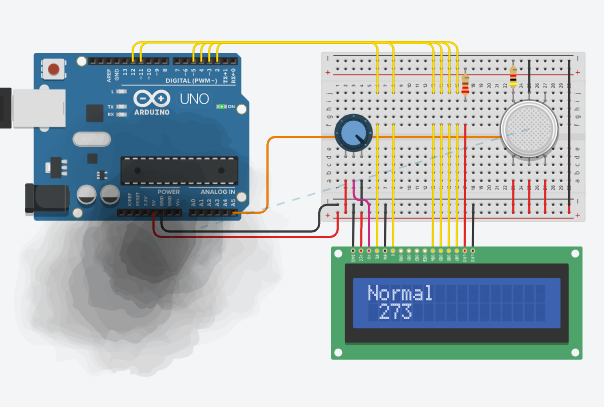
**Linh kiện:**

* Cảm biến khí Gas
* Màn hình LCD
* Điện trở
* Chiết áp
* Arduino Uno

**Code:**

|  |
| --- |
| *#include <LiquidCrystal.h>*  *LiquidCrystal lcd(12, 11, 5, 4, 3, 2);*  *int analogPin = A5;*  *int val = 0;*  *void setup() {*  *lcd.begin(16, 2);*  *// Print a message to the LCD.*  *lcd.print("Starting... ");*  *delay(500);*  *lcd.setCursor(1, 1);*  *lcd.print("Loading...");*  *delay(500);*  *Serial.begin(9600);*  *lcd.display();*  *}*  *void loop() {*  *lcd.setCursor(0,1);*  *delay(100);*  *val = analogRead(analogPin);*  *Serial.println(val);*  *lcd.clear();*  *if (val >0 && val <= 306){*  *lcd.print("Normal");*    *} else if (val > 307 && val <= 420){*  *lcd.print("Gas Detected!");*  *} else if (val > 421 && val <= 520){*  *lcd.print("Dense Gas!");*  *} else if (val > 521 && val <= 724){*  *lcd.print("Danger!");*  *}*  *lcd.setCursor(1, 1);*  *lcd.print(val);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **24. Thiết bị cảnh báo động đất sử dụng cảm biến độ nghiêng**

**Mô tả:** Khi cảm biến nghiêng một góc lớn hơn góc cho phép điện áp đầu ra tại chân A0 và A1 của cảm biến càng tăng lên. Bộ so sánh cùng với điện trở để điều chỉnh độ nhạy của cảm biến. Độ nghiêng được ghi lại và xuất ra màn hình LCD với 2 dòng là phương (xét dưới dạng không gian 2 chiều x và y) và độ nghiêng đo được.

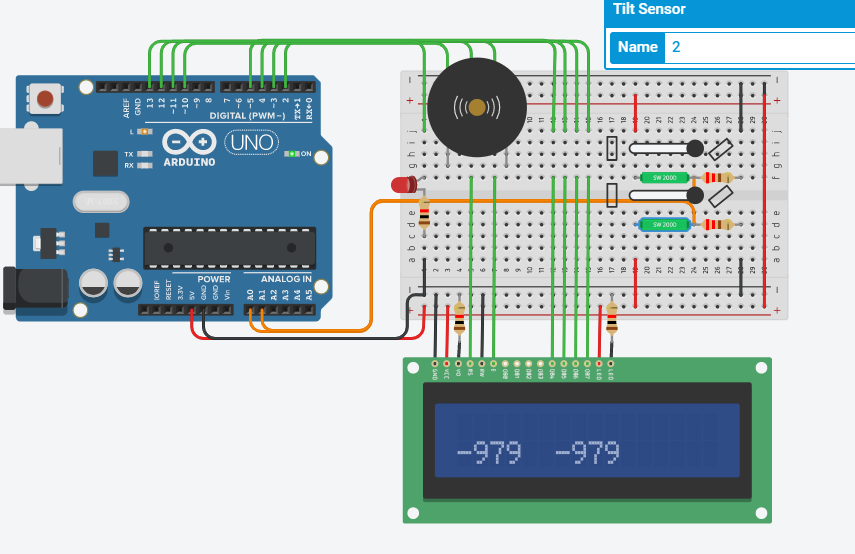
**Linh kiện:**

* Cảm biến Tilt ( Cảm biến độ nghiêng)
* Loa Piezo
* Đèn LED
* Điện trở
* Arduino Uno

**Code:**

|  |
| --- |
| *#include<LiquidCrystal.h>*  *LiquidCrystal lcd(12,11,5,4,3,2);*  *#define buzzer 10*  *#define led 13*  *#define x A0*  *#define y A1*  *//biến*  *int xsample=0;*  *int ysample=0;*  *int zsample=0;*  *long start;*  *int buz=0;*  */\*Macros\*/*  *#define samples 50*  *#define maxVal 20*  *#define minVal -20*  *#define buzTime 5000*  *void setup()*  *{*  *lcd.begin(16,2);*  *Serial.begin(9600);*  *delay(1000);*  *lcd.print("Earthquake");*  *lcd.setCursor(0,1);*  *lcd.print("Detector");*  *delay(2000);*  *lcd.clear();*  *lcd.print("Start");*  *lcd.setCursor(0,1);*  *lcd.print("Please wait...");*  *pinMode(buzzer, OUTPUT);*  *pinMode(led, OUTPUT);*  *buz=0;*  *digitalWrite(buzzer, buz);*  *digitalWrite(led, buz);*  *for(int i=0;i<samples;i++) // lấy mẫu để hiệu chuẩn*  *{*  *xsample+=analogRead(x);*  *ysample+=analogRead(y);*  *}*  *xsample/=samples; // lấy trung bình cho x*  *ysample/=samples; // lấy trung bình cho y*  *delay(3000);*  *lcd.clear();*  *lcd.print("Complete");*  *delay(1000);*  *lcd.clear();*  *lcd.print("Device Ready");*  *delay(1000);*  *lcd.clear();*  *lcd.print(" X Y");*  *}*  *void loop()*  *{*  *int value1=analogRead(x); //đọc x ra*  *int value2=analogRead(y); //đọc y ra*  *int xValue=xsample-value1; // tìm kiếm sự thay đổi của x*  *int yValue=ysample-value2; // tìm kiếm sự thay đổi của y*  *// hiển thị thay đổi các giá trị trục x, y trên LCD*  *lcd.setCursor(0,1);*  *lcd.print(xValue);*  *lcd.setCursor(6,1);*  *lcd.print(yValue);*  *delay(100);*  *lcd.clear();*  *// so sánh thay đổi với các giới hạn được xác định trước*  *if(xValue < minVal || xValue > maxVal || yValue < minVal || yValue > maxVal)*  *{*  *if(buz == 0)*  *start=millis(); // hẹn giờ bắt đầu*  *buz=1; // buzzer / leb báo hiệu được kích hoạt*  *}*  *else*  *if(buz == 1) // buzzer được kích hoạt rồi cảnh báo động đất*  *{*  *lcd.setCursor(0,0);*  *lcd.print("Earthquake Alert");*  *if(millis()>= start+buzTime)*  *buz=0;*  *}*  *else*  *{*  *lcd.clear();*  *lcd.print("X Y");*  *}*  *digitalWrite(buzzer, buz); // lệnh bật và tắt buzzer*  *digitalWrite(led, buz); // lệnh bật và tắt LED*  *// gửi các giá trị để xử lý*  *Serial.print("x=");*  *Serial.println(xValue);*  *Serial.print("y=");*  *Serial.println(yValue);*  *Serial.println(" $");*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **25. Sử dụng quang điện trở để điều khiển đèn LED**

**Mô tả:** Khi quang điện trở không thu được ánh sáng điện áp đầu ra tại chân A5 tăng lên. Dựa vào giá trị đọc được ở A5 từ đó điều khiển đèn bật,tắt.

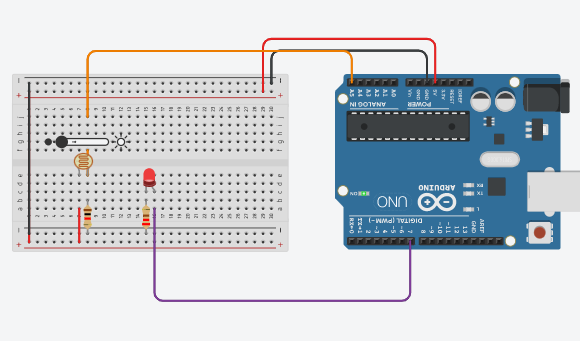
**Linh kiện:**

* Quang điện trở
* Điện trở
* Đèn LED
* Arduino Uno

**Code:**

|  |
| --- |
| *void setup()*  *{*  *pinMode(7, OUTPUT);*  *Serial.begin(9600);*  *}*  *void loop()*  *{*  *Serial.println(analogRead(5));*  *if(analogRead(5) < 500){*  *digitalWrite(7, HIGH);*  *} else {*  *digitalWrite(7, LOW);*  *}*  *delay(100);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



## **26. Sử dụng Board micro:bit điều khiển đèn sáng tuần tự**

**Mô tả:** Board micro:bit có kích thước nhỏ (4 x 5 cm), gồm 1 ma trận led 5 x 5 để hiển thị, 2 nút bấm lập trình được, 1 nút reset, 3 chân in/out, cùng với một vài led để báo hiệu. Sử dụng các chân 1,2,3 tương ứng với các đèn để điều khiển bật/tắt tuần tự.

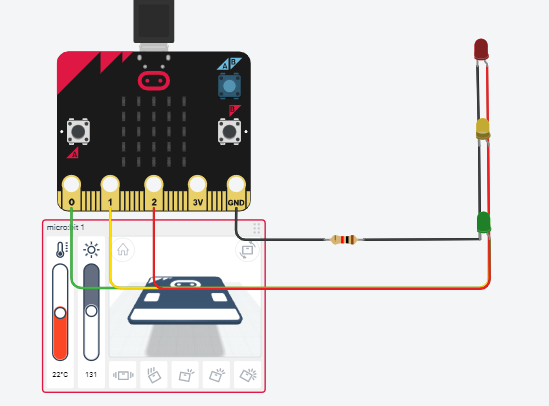
**Linh kiện:**

* Micro Servo (dạng motor góc quay nằm trong khoảng bất kì từ 0o - 180o)
* Điện trở 1k Ω
* Cảm biến Flex (cảm biến cong)
* Arduino Uno

**Blocks:**

|  |
| --- |
|  |

**Sơ đồ thiết kế với Tinkercad :**



# **II. Các bài thực hành với STM32**

## **1. Sử dụng STM32 điều khiển 4 LED**

**Mô tả:** Đèn LED được nối vào các chân PD12 đến PD15 do đó khi thực hiện thay đổi tín hiệu giữa các chân sẽ khiến cho đèn LED bật/tắt tuần tự.

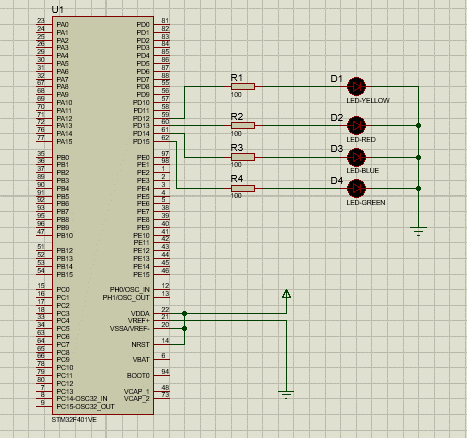
**Linh kiện:**

* STM32F407VG
* Điện trở 100 Ω
* Cảm biến Flex (cảm biến cong)
* Arduino Uno

**Code chính:**

|  |
| --- |
| *static void MX\_GPIO\_Init(void)*  *{*  *GPIO\_InitTypeDef GPIO\_InitStruct = {0};*  */\* GPIO Ports Clock Enable \*/*  *\_\_HAL\_RCC\_GPIOH\_CLK\_ENABLE();*  *\_\_HAL\_RCC\_GPIOD\_CLK\_ENABLE();*  */\*Configure GPIO pin Output Level \*/*  *HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15, GPIO\_PIN\_RESET);*  */\*Configure GPIO pins : PD12 PD13 PD14 PD15 \*/*  *GPIO\_InitStruct.Pin = GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15;*  *GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*  *GPIO\_InitStruct.Pull = GPIO\_NOPULL;*  *GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_HIGH;*  *HAL\_GPIO\_Init(GPIOD, &GPIO\_InitStruct);*  *}* |

**Sơ đồ thiết kế với Tinkercad :**



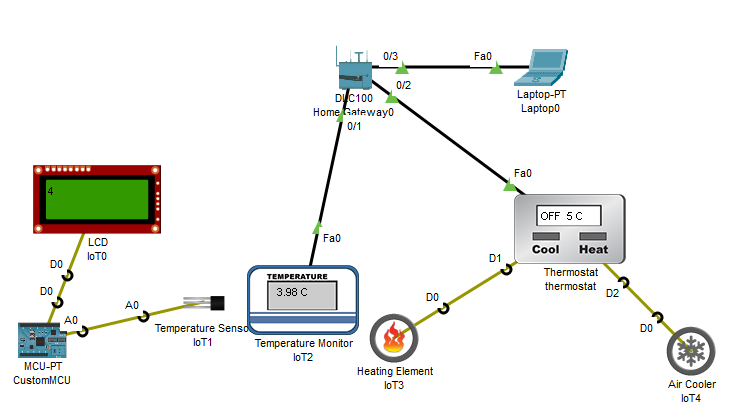
# **III. Các chương trình IoT**

## **1.Đo nhiệt độ bằng cảm biến sau đó hiển thị trên LCD có sử dụng Arduino**

**Mô tả:** Đo nhiệt độ môi trường rồi hiển thị ra LCD

**Thiết bị:** Máy tính, Home Gateway, Thiết bị điều chỉnh nhiệt độ, Thiết bị giám sát nhiệt, Arduino, Cảm biến nhiệt, Màn hình LCD.

**Mô hình:**



**Code Arduino - Python:**

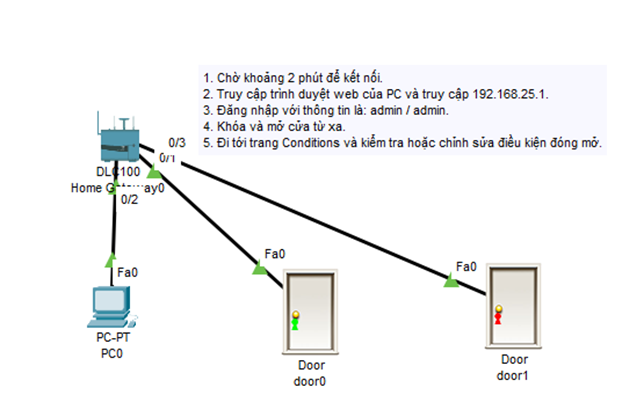
|  |
| --- |
| *from gpio import \**  *from time import \**  *def inputHandler():*  *value = (((analogRead(A0) - 0) \* (100 - -100)) / (1023 - 0)) + -100*  *customWrite(0, value)*  *def main():*  *add\_event\_detect(A0, inputHandler)*  *while True:*  *delay(1000)*  *if \_\_name\_\_ == "\_\_main\_\_":*  *main()* |

## **2.Đóng mở cửa với IoT**

**Mô tả:** Sử dụng ứng dụng của IoT cho việc đóng mở cửa. Dựa vào trình duyệt web của PC và truy cập 192.168.25.1 để đưa ra câu lệnh đóng/ mở cửa.

**Thiết bị:** Máy tính, Home Gateway

**Mô hình:**

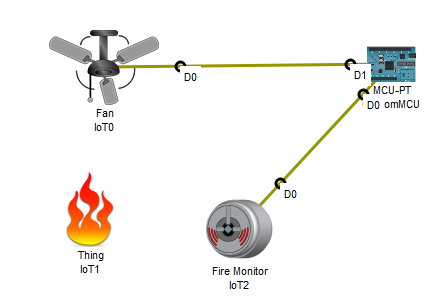
****

## **3.Giám sát nhiệt độ và ra lệnh có điền kiện**

**Mô tả:** Dùng thiết bị giám sát nhiệt độ, nếu nhiệt độ cao thì sẽ khởi động quạt.

**Thiết bị:** Quạt, Giám sát nhiệt độ, Arduino.

**Mô hình:**



**Code Arduino – Python:**

|  |
| --- |
| *from gpio import \**  *from time import \**  *def handleSensorData():*  *value = digitalRead(0)*  *if value == 0:*  *customWrite(1, '0')*  *else:*  *customWrite(1, '1')*  *def main():*  *add\_event\_detect(0, handleSensorData)*  *while True:*  *delay(1000)*  *if \_\_name\_\_ == "\_\_main\_\_":*  *main()* |

## **4.Bật/tắt thiết bị từ xa thông qua Server**

**Mô tả:** Điều khiển thiết bị từ xa sử dụng máy tính ra lệnh, cấu hình IP thông qua Server.

**Linh kiện:** Server, Máy tính hoặc thiết bị kết nối khác, Thiết bị cần điều khiển.

**Mô hình:**

