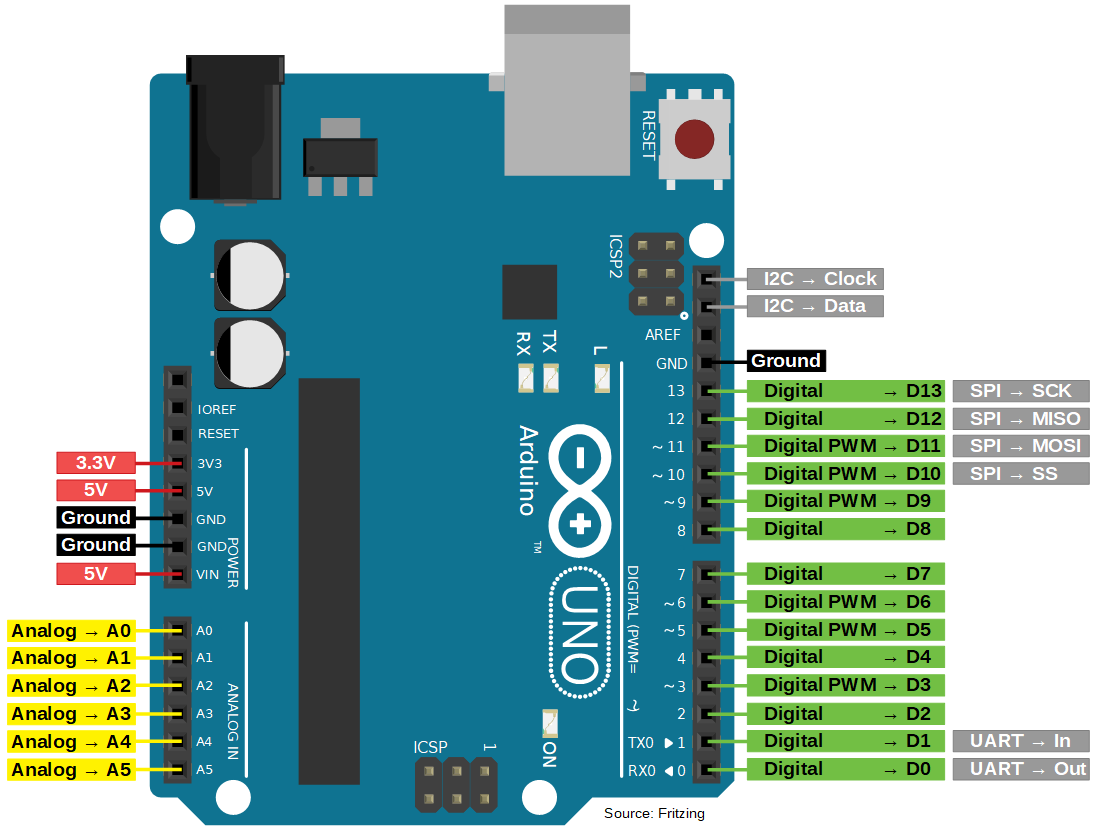
1. **Tổng quan:**

* Vỗ tay hai nhịp để thay đổi các chế độ sáng của LED bằng cảm biến âm thanh.

1. Phần cứng:

|  |  |  |
| --- | --- | --- |
| STT | Tên | SL |
| 1 | Arduino Uno R3 | 1 |
| 2 | Cảm biến âm thanh CN07(3 chân) | 1 |
| 3 | Biến trở Volume (1K) | 1 |
| 4 | LED đỏ (5mm) | 6 |
| 5 | Trở cắm (R220 Ohm ) | 6 |

*Một vài thông số của Arduino UNO R3*



|  |  |
| --- | --- |
| Vi điều khiển | ATmega328 họ 8bit |
| Điện áp hoạt động | 5V DC (chỉ được cấp qua cổng USB) |
| Tần số hoạt động | 16 MHz |
| Dòng tiêu thụ | khoảng 30mA |
| Điện áp vào khuyên dùng | 7-12V DC |
| Điện áp vào giới hạn | 6-20V DC |
| Số chân Digital I/O | 14 (6 chân hardware PWM) |
| Số chân Analog | 6 (độ phân giải 10bit) |
| Dòng tối đa trên mỗi chân I/O | 30 mA |
| Dòng ra tối đa (5V) | 500 mA |
| Dòng ra tối đa (3.3V) | 50 mA |
| Bộ nhớ flash | 32 KB (ATmega328) với 0.5KB dùng bởi bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |

*Cảm biến CN07*

- Nó được sử dụng để phát hiện âm thanh, tiếng động xung quang... Từ đó báo tín hiệu về và xuất ra chân tín hiệu. Chúng ta có thể điều chỉnh độ nhạy cho cảm biến thông qua 1 con biến trở. Điện áp có thể hoạt động là từ 3V3 - 5V. Tín hiệu ra là dạng digital (Nhưng có loại cho ra cả tín hiệu analog-Âm lượng).

* VCC: nguồn 5V
* GND: Mass
* OUT: đầu ra tín hiệu số (mức cao hoặc mức thấp)



*Biến trở Volume*

- Khi ta tăng/giảm giá trị của biến trở bằng cách xoay con chạy thì sẽ làm tăng/giảm dòng điện đi qua.

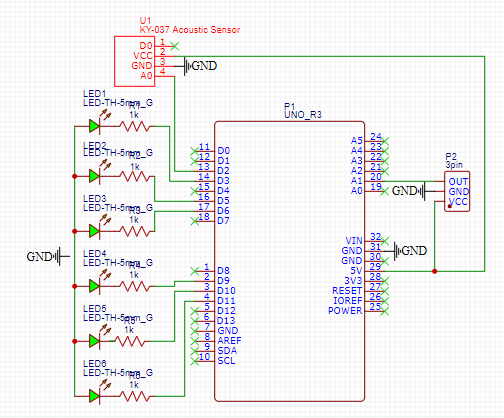
* Chân trái: Mass
* Chân giữa: đầu ra Analog
* Chân phải: nguồn 5V

1. Phần mềm:

* Arduino 1.8.12: viết code, nạp code cho Arduino Uno R3.
* Proteus 8: vẽ sơ đồ nguyên lý, PCB.

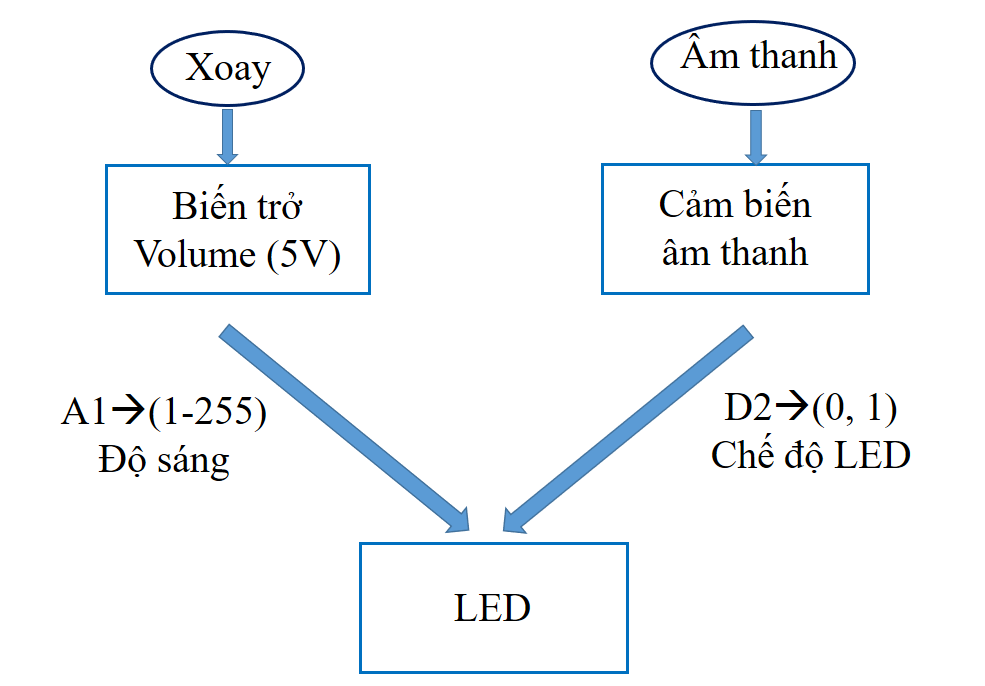
1. **Nguyên lý hoặt động**
2. Sở đồ nguyên lý

* Các chân Digital PWM (D3, D5, D6, D9, D10, D11): phát xung điều khiển 6 bóng LED.
* Chân Digital D2: nhận tín hiệu Digital từ Module cảm biến âm thanh.
* Chân Analog A1: nhận tín hiệu Analog từ biến trở Volume.
* Cấp nguồn 5V cho sensor.



Cảm biến âm thanh ghi nhận âm thanh (tiếng vỗ tay) chuyển thành tín hiệu Digital được chân D2 ghi lại (0: có âm thanh, 1: không có âm thanh, quy ước có thể thay đổi trong lập trình), biến trở trên cảm biến giúp điều chỉnh độ nhạy.

Biến trở Volume: xoay con chạy tăng/giảm dòng điện chân OUT trong khoảng 0-5V được chân A1 ghi lại tương ứng với thang đo 0-255 trong lập trình độ sáng đèn LED, để tránh LED khồng bị tắt khi xoay giảm hết mức ta điều chỉnh về thang đo 1-255.



1. Phần lập trình

#define volumregister A1

#define sensor 2

int clap = 0;

int status\_sensor;

int val=0;

int j=-1;

int j1=-1;

byte ledPin[] = {3,5,6,9,10,11};

byte pinCount;

long detection\_range\_start = 0;

long detection\_range = 0;

long sta;

boolean mode1= false;

boolean mode2= false;

boolean s=true;

char k=0;

void setup() {

sta=millis();

pinMode(volumregister,INPUT);

pinMode(sensor,INPUT);

pinCount = sizeof(ledPin);

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

{

pinMode(ledPin[i],OUTPUT);

}

}

void loop()

{

if (digitalRead(sensor) == 0)

{

if (clap == 0)

{

detection\_range\_start = detection\_range = millis();

clap++;

}

else if (clap > 0 && millis()-detection\_range >= 50)

{

detection\_range = millis();

clap++;

}

}

if (millis()-detection\_range\_start >= 400)

{

if (clap == 2)

{

if (!mode1)

{

if (!mode2)

{

mode2=true;

k=1;

}

else if (mode2) {

mode1 = true;

mode2=false;

k=2;

j=-1;

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

analogWrite(ledPin[pinCount-i],0);

}

}

}

else if (mode1)

{

if (!mode2) {

mode2=true;

k=3;

j1=-1;

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

analogWrite(ledPin[pinCount-i],0);

}

}

else if (mode2){

mode1 = false;

mode2=false;

k=4;

}

}

}

clap = 0;

}

val=map(analogRead(volumregister),0,1023,1,255);

if (k==1){

if (millis()-sta>=500){

if (s){

analogWrite(3,0);analogWrite(5,val);

analogWrite(6,0);analogWrite(9,val);

analogWrite(10,0);analogWrite(11,val);

s=false;

sta=millis();

}

else if (!s){

analogWrite(3,val);analogWrite(5,0);

analogWrite(6,val);analogWrite(9,0);

analogWrite(10,val);analogWrite(11,0);

sta=millis();

s=true;

}

}

}

else if (k==2){

if (millis()-sta>=25){

if (s){

j++;

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

analogWrite(ledPin[i],255-j);

if (255-j==0){s=false;j=-1;for (int i=0;i<=2;i++){analogWrite(ledPin[i],0);}}

}

sta=millis();

}

else if (!s){

j++;

for (int i=3;i<=5;i++){

analogWrite(ledPin[i],255-j);

if (255-j==0){s=true;j=-1;for (int i=3;i<=5;i++){analogWrite(ledPin[i],0);}}

}

sta=millis();

}

}

}

else if (k==3){

if (millis()-sta>=200){

if (s){

j1++;

analogWrite(ledPin[j1],val);

if (j1==6){j1=-1;}

s=false;

sta=millis();

}

else if (!s){

//j1;

analogWrite(ledPin[j1],0);

if(j1==5){j1=-1;}

s=true;

sta=millis();

}

}

}

else if (k==4) {

if (millis()-sta>=500){

if (s){

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

analogWrite(ledPin[pinCount-i],val);

delay(35);

}

s=false;

sta=millis();

}

else if (!s){

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

analogWrite(ledPin[pinCount-i],0);

delay(35);

}

s=true ;

sta=millis();

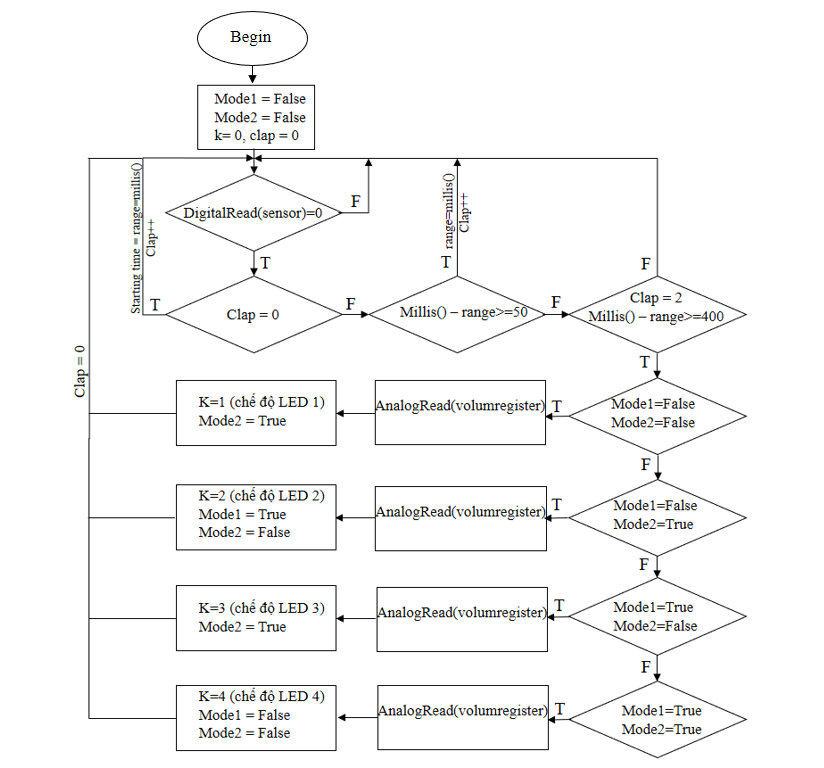
}

}

}

}

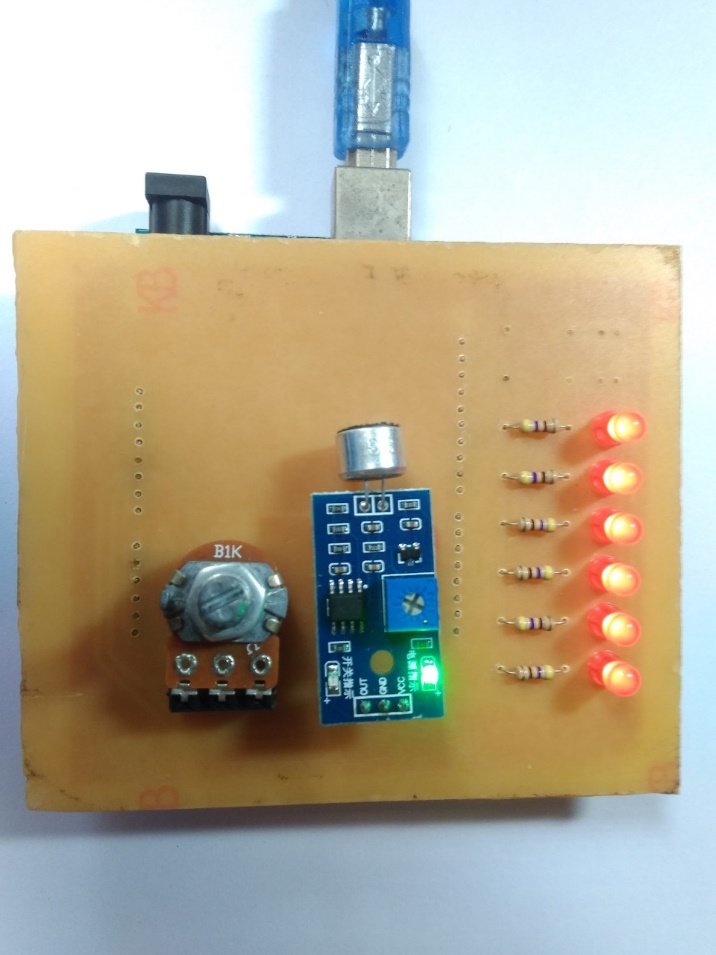
Lưu đồ thuật toán:



1. **Vận hành**

* Chuyển chế độ LED bằng cách vỗ tay 2 lần liên tiếp, vặn núm xoay trên biến trở để tăng/giảm độ sáng.

Một số hình ảnh vận hành:



**Kết luận:** nhóm đã hoàn thành mạch, đầy đủ 4 chế độ LED, mạch hoạt động tốt và nhạy.

* Khó khăn:
* Về việc chọn cảm biến âm thanh, có một số loại không nhạy (vd: CN07 4 chân), dễ bị nhiễu và dễ hỏng (không sáng LED trạng thái, không phản hồi tín hiệu).
* Yêu cầu cảm biến đọc tín hiệu liên tục và độ sáng của LED phải phản hồi theo giá trị của biến trở ngay lập tức.
* Khắc phục:
* Thay cảm biến CN07 loại 4 chân bằng CN07 loại 3 chân giúp hoạt động nhạy hơn, phản hồi tính hiệu tốt.
* Trong lập trình loại bỏ những hàm lặp nhiều lần (vd: hàm for, delay(),…) thay bằng hàm điều kiện if, hàm thời gian millis() giúp cảm biến đọc dữ liệu và phản hồi liên tục🡪 LED hoạt động theo yêu cầu không bị trễ.