

# Persistence of Vision

## Autor

[Alice-Maria PÎRVULESCU \(101523\)](#)

## Descriere

Persistența viziunii (**POV**) este o parte curioasă a sistemului vizual uman.

Acesta reprezintă efectul prin care percepția unei imagini persistă după ce lumina a încetat să pătrundă în ochi. Este motivul pentru care o elice care se învâрте apare ca un disc și de ce o scânteie aprinsă pare să lase o urmă în aer.

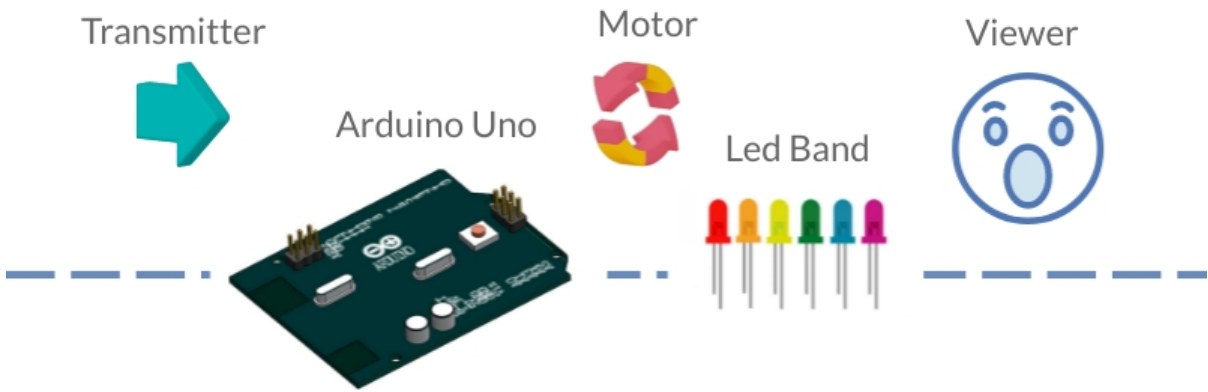
Este, de asemenea, utilizat în mod obișnuit ca tehnologie de afișare, unde o serie de LED-uri intermitente pot fi utilizate pentru a crea mesaje care par să plutească în aer.

Cu ajutorul unei benzi de leduri, un motor care le va roti și faptul că imaginea persista timp de 1/25 secunde pe retina după momentul dispariției ei, voi crea iluzia **persistenței unui cuvânt**.

## Componente

\* Arduino Uno \* DC Motor \* +9V battery \* Led Band \* 1KΩ resistors

## Idee



# Block scheme

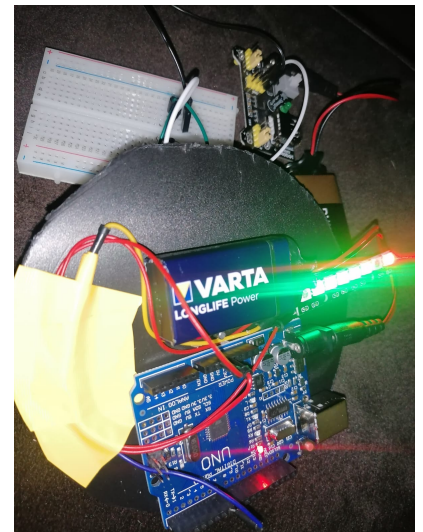


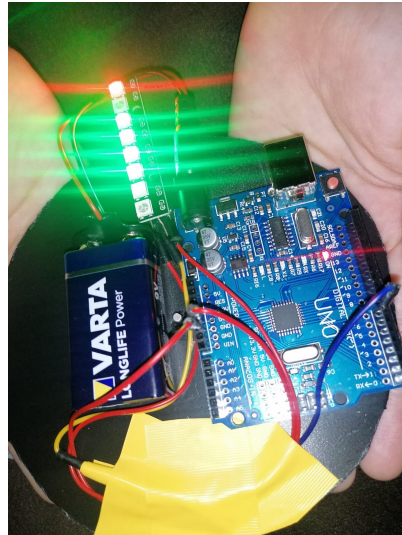
Cum functioneaza crearea literelor

## Schema



## Schema fizica





## Cod

```
#include <FastLED.h>

#define LED_PIN    7
#define NUM_LEDS  8

int ledPin = 13;
int state = 0;

CRGB leds[NUM_LEDS];

double delayTime = 1;
double charBreak = 2.5;

void setup(){
  FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS);
  pinMode(ledPin, OUTPUT);
}

int a[] = {1, 6, 26, 6, 1};
int b[] = {31, 21, 21, 10, 0};
int c[] = {14, 17, 17, 10, 0};
int d[] = {31, 17, 17, 14, 0};
int e[] = {31, 21, 21, 17, 0};
int f[] = {31, 20, 20, 16, 0};
int g[] = {14, 17, 19, 10, 0};
int h[] = {31, 4, 4, 4, 31};
int i[] = {0, 17, 31, 17, 0};
int j[] = {0, 17, 30, 16, 0};
int k[] = {31, 4, 10, 17, 0};
int l[] = {31, 1, 1, 1, 0};
int m[] = {31, 12, 3, 12, 31};
```

```
int n[] = {31, 12, 3, 31, 0};
int o[] = {14, 17, 17, 14, 0};
int p[] = {31, 20, 20, 8, 0};
int qQ[] = {14, 17, 19, 14, 2};
int r[] = {31, 20, 22, 9, 0};
int s[] = {8, 21, 21, 2, 0};
int t[] = {16, 16, 31, 16, 16};
int u[] = {30, 1, 1, 30, 0};
int v[] = {24, 6, 1, 6, 24};
int w[] = {28, 3, 12, 3, 28};
int x[] = {17, 10, 4, 10, 17};
int y[] = {17, 10, 4, 8, 16};
int z[] = {19, 21, 21, 25, 0};

int eos[] = {0, 1, 0, 0, 0};
int excl[] = {0, 29, 0, 0, 0};
int ques[] = {8, 19, 20, 8, 0};

void displayLine(int line){
    int myline;
    myline = line;
    if (myline>=16) {leds[1] = CRGB(0, 0, 255); myline-=16;}
    else {leds[1] = CRGB::Black;}
    if (myline>=8) {leds[2] = CRGB(0, 0, 255); myline-=8;}
    else {leds[2] = CRGB::Black;}
    if (myline>=4) {leds[3] = CRGB(0, 0, 255); myline-=4;}
    else {leds[3] = CRGB::Black;}
    if (myline>=2) {leds[4] = CRGB(0, 0, 255); myline-=2;}
    else {leds[4] = CRGB::Black;}
    if (myline>=1) {leds[5] = CRGB(0, 0, 255); myline-=1;}
    else {leds[5] = CRGB::Black;}
    leds[7] = CRGB(255, 0, 0);
    FastLED.show();
}

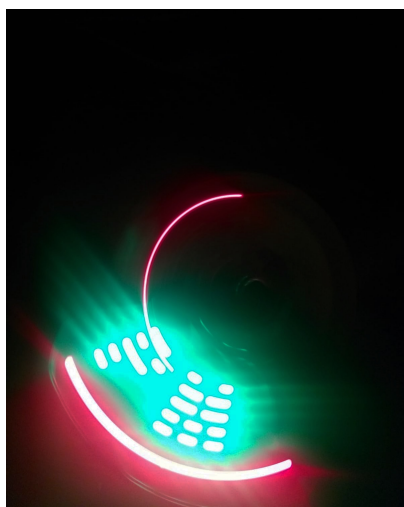
void displayChar(char ch){
    if (ch == 'a'){for (int i = 0; i <5;
i++){displayLine(a[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'b'){for (int i = 0; i <5;
i++){displayLine(b[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'c'){for (int i = 0; i <5;
i++){displayLine(c[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'd'){for (int i = 0; i <5;
i++){displayLine(d[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'e'){for (int i = 0; i <5;
i++){displayLine(e[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'f'){for (int i = 0; i <5;
i++){displayLine(f[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'g'){for (int i = 0; i <5;
i++){displayLine(g[i]);delay(delayTime);}displayLine(0);}
```

```
    if (ch == 'h'){for (int i = 0; i <5;
i++){displayLine(h[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'i'){for (int it = 0; it <5;
it++){displayLine(i[it]);delay(delayTime);}displayLine(0);}
    if (ch == 'j'){for (int i = 0; i <5;
i++){displayLine(j[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'k'){for (int i = 0; i <5;
i++){displayLine(k[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'l'){for (int i = 0; i <5;
i++){displayLine(l[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'm'){for (int i = 0; i <5;
i++){displayLine(m[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'n'){for (int i = 0; i <5;
i++){displayLine(n[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'o'){for (int i = 0; i <5;
i++){displayLine(o[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'p'){for (int i = 0; i <5;
i++){displayLine(p[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'q'){for (int i = 0; i <5;
i++){displayLine(q[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'r'){for (int i = 0; i <5;
i++){displayLine(r[i]);delay(delayTime);}displayLine(0);}
    if (ch == 's'){for (int i = 0; i <5;
i++){displayLine(s[i]);delay(delayTime);}displayLine(0);}
    if (ch == 't'){for (int i = 0; i <5;
i++){displayLine(t[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'u'){for (int i = 0; i <5;
i++){displayLine(u[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'v'){for (int i = 0; i <5;
i++){displayLine(v[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'w'){for (int i = 0; i <5;
i++){displayLine(w[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'x'){for (int i = 0; i <5;
i++){displayLine(x[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'y'){for (int i = 0; i <5;
i++){displayLine(y[i]);delay(delayTime);}displayLine(0);}
    if (ch == 'z'){for (int i = 0; i <5;
i++){displayLine(z[i]);delay(delayTime);}displayLine(0);}
    if (ch == '!'){for (int i = 0; i <5;
i++){displayLine(excl[i]);delay(delayTime);}displayLine(0);}
    if (ch == '?'){for (int i = 0; i <5;
i++){displayLine(ques[i]);delay(delayTime);}displayLine(0);}
    if (ch == '.'){for (int i = 0; i <5;
i++){displayLine(eos[i]);delay(delayTime);}displayLine(0);}
    delay(charBreak);
}

void displayString(char* s){
for (int i = 0; i<=strlen(s); i++){
displayChar(s[i]);
}
```

```
}  
  
void loop()  
{  
  displayString("hello");  
}
```

## Rezultat final



Persistența imaginii este proprietatea ochiului nostru de a vedea în continuare (sau iluzia de a vedea) o formă a ceva, pentru o fracțiune de secundă chiar și după ce obiectul a fost mutat. Proiectul se bazează pe vederea umană. Nu am reușit să fac videoclip deoarece această iluzie nu se păstrează, iar pentru a face pozele am folosit opțiunea de fotografie continuă a camerei.

## Mai multe despre Persistence of Vision

[https://en.wikipedia.org/wiki/Persistence\\_of\\_vision](https://en.wikipedia.org/wiki/Persistence_of_vision)

<https://hackaday.com/2019/10/29/the-basics-of-persistence-of-vision-projects/>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/abirlica/perceptionofvision>



Last update: **2021/06/04 16:14**