

Autor: Anghel Bogdan Gabriel

Grupa: 333CC

## Introducere

Un display persistence of vision ce prezinta un ceas caruia i se poate seta o alarma.

## Descriere generala

Se conecteaza un board cu 8/16 LEDuri la un Arduino UNO alaturi de un senzor. Prin rotirea suportului pe care sunt montate ledurile cu ajutorul motorului DC, se va obtine un text reprezentand ora curenta care va parea persistent ochiului uman. Acesta va avea un buzzer conectat care va suna o alarma cand se ajunge la o ora prestabilita.

## Schema Bloc



## Schema Electrica



## Hardware Design

### Lista de Piese

- ARDUINO UNO
- 8/16 LEDuri
- Motor
- Baterii 9V
- Suport
- Placa PCB
- Rezistente
- Fire
- Switch
- Buzzer
- Senzor IR

## Asamblare

1. am pus 8 leduri pe pcb, am legat catodul in comun si l-am conectat la GND, iar la anod cate o rezistenta;
2. am lipit pinii de placuta utilizand un pistol de lipit;
3. am lipit jumper cables M/M de rezistente si conectat la placuta Arduino la pinii 2-9;
4. am conectat senzorul IR la placuta la pinii 10-11;
5. am conectat buzzerul la placuta la pinul 13;
6. am conectat o baterie 9V la o mufa de alimentare pentru a o putea conecta la placuta Arduino;
7. am montat pe o placa de gips-carton subtire (spoiler: o sa fie prea grea) componentele mentionate anterior si le-am lipit cu silicon;
8. am atasat suportul pe un ventilator pentru a invarti placa.

## Software Design

Codul a fost scris in Arduino IDE. M-am inspirat din versiuni gasite pe internet, modificand pentru a se potrivi porturile, senzorul, sincronizarea, caracterele. Am mapat caracterele, fiecare avand o secventa

de 5 faze:

```
int NUMBER9[]={1,1,1,1,0,0,0,1, 1,0,0,1,0,0,0,1, 1,0,0,1,0,0,0,1,
1,0,0,1,0,0,0,1, 1,1,1,1,1,1,1,1};
int NUMBER8[]={0,1,1,0,1,1,1,0, 1,0,0,1,0,0,0,1, 1,0,0,1,0,0,0,1,
1,0,0,1,0,0,0,1, 0,1,1,0,1,1,1,0};
int NUMBER7[]={1,0,0,0,0,0,0,0, 1,0,0,0,1,0,0,0, 1,0,0,0,1,0,0,0,
1,0,0,1,1,1,1,1, 1,1,1,0,1,0,0,0};
int NUMBER6[]={1,1,1,1,1,1,1,1, 1,0,0,0,1,0,0,1, 1,0,0,0,1,0,0,1,
1,0,0,0,1,0,0,1, 1,0,0,0,1,1,1,1};
int NUMBER5[]={1,1,1,1,1,0,0,1, 1,0,0,0,1,0,0,1, 1,0,0,0,1,0,0,1,
1,0,0,0,1,0,0,1, 1,0,0,0,1,1,1,1};
. . .
. . .
```

Funcția printLetter afișează litera curentă cu ajutorul ledurilor

```
void setup()
{
    Serial.begin(9600);
    pinMode(IR_pin,INPUT);
    for( int i = 2; i<10 ;i++ )
        {
            pinMode(i, OUTPUT);
        }

    letterSpace =4;
    delayTime =1;
}

void printLetter(int letter[])
{
    int y;
    for (y=0; y<8; y++)
        {
            digitalWrite(y+2, letter[y]);
        }
    delay(delayTime);
    for (y=0; y<8; y++)
        {
            digitalWrite(y+2, letter[y+8]);
        }
    delay(delayTime);
    for (y=0; y<8; y++)
        {
            digitalWrite(y+2, letter[y+16]);
        }
    delay(delayTime);
    for(y = 0; y<8; y++) {
        digitalWrite(y+2, letter[y+24]);
    }
```

```
    }
    delay(delayTime);
    for(y = 0; y<8; y++) {
        digitalWrite(y+2, letter[y+32]);
    }
    delay(delayTime);
    for (y=0; y<8; y++)
    {
        digitalWrite(y+2, 0);
    }
    delay(letterSpace);

    pinMode(13, OUTPUT);

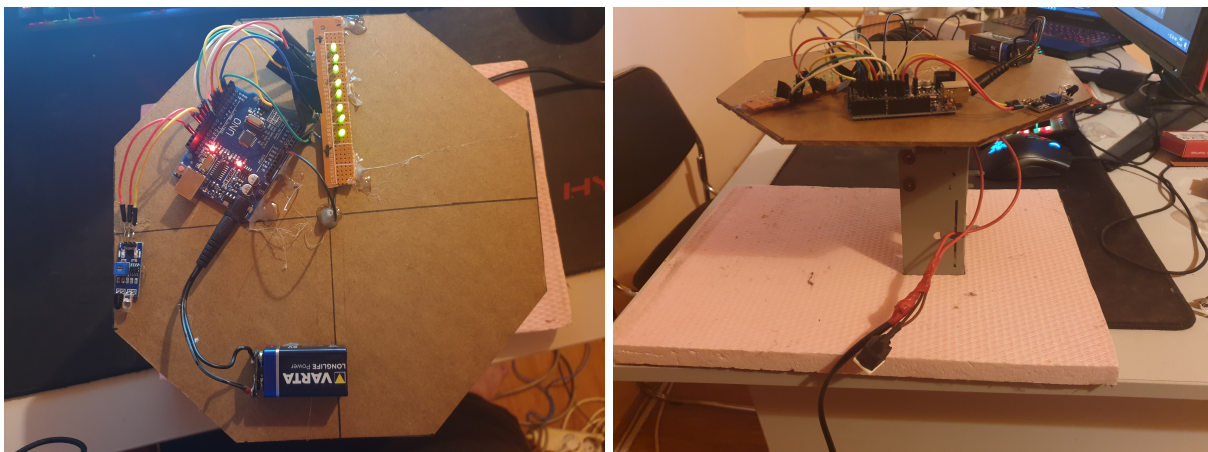
void loop()
{
    if(digitalRead(IR_pin)==LOW)
    {
        printLetter (P);
        printLetter (0);
        printLetter (V);
    }
}
```

### Codul ca txt

#### Rezultate Obtinute

Acesta este aspectul final al proiectului. Din pacate nu functioneaza efectiv, intrucat motorul este prea slab pentru a putea invarti suportul suficient de repede. Voi incerca sa inlocuiesc fie suportul fie motorul pentru a-l face functional.

Datorita lipsei unui modul RTC, nu am putut realiza functionalitatea initial dorita, aceea de a avea un ceas POV cu alarma. Buzzerul functioneaza la trigger (hardcoded), dar neavand un timp setat pentru a reprezenta o alarma acesta nu se activeaza automat.



## Demo proiect

### Concluzii

A fost un proiect interesant si regret ca nu i-am acordat mai multa atentie si timp. Dificultatea nu a fost una mare, dar lipsa componentelor adecvate (motor mai puternic / placuta mai usoare si modulul RTC) m-au impiedicat din ducerea proiectului la sfarsit. Voi incerca sa finalizez proiectul dupa terminarea examenelor pentru a avea satisfactia de proiect terminat.

### Jurnal

25.04 Alegere tema proiect

11.05 Comanda piese

31.05 Lucrarea la partea hardware

01.06 Finalizarea hardware si software

### Bibliografie/Resurse

### Video asamblare

[Exemplu montaj si lista componente](#)

[PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2021/avaduva/pov\\_alarma](http://ocw.cs.pub.ro/courses/pm/prj2021/avaduva/pov_alarma)



Last update: **2021/06/02 08:45**