

# Piano Player

## Introducere

Prezentarea pe scurt a proiectului:

Este un robot care poate interactiona cu jocul de mobile 'Piano Tiles'.  
Va recunoaste diferenta dintre culorile patratelor care apar pe ecran si va apasa pe cele negre.  
Am jucat destul de mult acest joc si as vrea sa vad daca un robot ar putea depasi limitele unui om.

## Descriere generală

Proiectul reprezinta un robot care poate juca popularul joc Piano Tiles.

Robotul va citi valori de segmente din ecran si in functie de culoarea pe care o citeste pe fiecare segment va simula sau nu o atingere a ecranului touchscreen.

## Hardware Design

Piese necesare:

Arduino UNO  
LDR (photoresistor)  
Power Relays  
Jumper Wires  
Rezistors(22kohm)  
Breadboard

Schema Bloc:



Hardware Design:

Proiectul poate fi impartit in 2 segmente: achizitia de date (photorezistorii) si simularea atingerii touch screen (prin relays).

Photorezistorii o sa fie positionati astfel incat sa poata citi valorile de pe ecran;

Pentru negru, valoarea data la input va fi mai mica decat pentru alb, photorezistorul va interpreta culoarea negru ca lipsa luminii.

Atunci cand un LDR citeste culoarea negru, acesta va declansa o mutare a relayului, mai exact inchiderea switchului.

Ecranele touchscreen sunt ecrane capacitive, deci atunci cand noi atingem ecranul ne comportam ca o impamantare pt circuit. Voi incerca sa simulez un comportament similar prin folosirea unor switchuri care se conecteaza la ground cand vrem sa "apasam".

Voi atasa in continuare o imagine cu schema proiectului in ThinkerCad:



Schematic:



In proiectul fizic se vor inlocui multimeter-urile cu fire care au un material conductiv la capat, care vor fi pozitionate pe ecranul telefonului, dar in ThinkerCad folosesc multimeters pentru a ma asigura ca relayurile comuta intre 5V si ground (~0).

## Software Design

Codul folosit este unul simplu, care trebuie doar sa citeasca cate un input de la fiecare photorezistor si in functie de valoarea acestora va schimba valoarea unui pin conectat la un relay intre HIGH si LOW.

Codul initial, testat in ThinkerCad, arata astfel:

```
int photo0 = A0;
int photo1 = A1;
int photo2 = A2;
int photo3 = A3;

int relay0 = 10;
int relay1 = 11;
int relay2 = 12;
int relay3 = 13;

int delay_close_1 = 50;
int delay_close_2 = 50;

int delay_far = 100;
```

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(photo0, INPUT);
  pinMode(photo1, INPUT);
  pinMode(photo2, INPUT);
  pinMode(photo3, INPUT);

  pinMode(relay0, OUTPUT);
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  pinMode(relay3, OUTPUT);

}

void loop() {

  if (analogRead(photo0) < 700) {
    delay(delay_far);
    digitalWrite(relay0, HIGH);
    delay(delay_close_1);
    digitalWrite(relay0, LOW);
    delay(delay_close_2);
  } else if (analogRead(photo1) < 700) {
    digitalWrite(relay1, HIGH);
    delay(delay_close_1);
    digitalWrite(relay1, LOW);
    delay(delay_close_2);
  } else if (analogRead(photo2) < 700) {
    delay(delay_far);
    digitalWrite(relay2, HIGH);
    delay(delay_close_1);
    digitalWrite(relay2, LOW);
    delay(delay_close_2);
  } else if (analogRead(photo3) < 700) {
    digitalWrite(relay3, HIGH);
    delay(delay_close_1);
    digitalWrite(relay3, LOW);
    delay(delay_code_2);
  }
}
```

Photorezistorii sunt atasati catre porturile A0, A1, A2, A3 si trimit pe acestea valori semnificand intensitatea luminoasa citita. Atunci cand eranul afiseaza culoarea negru (o clapa), valoarea returnata de LDR este mai mica de 700.

O valoare mai mica de 700 rezulta in comutarea unui releu la HIGH si apoi la LOW pentru a simula atingere.

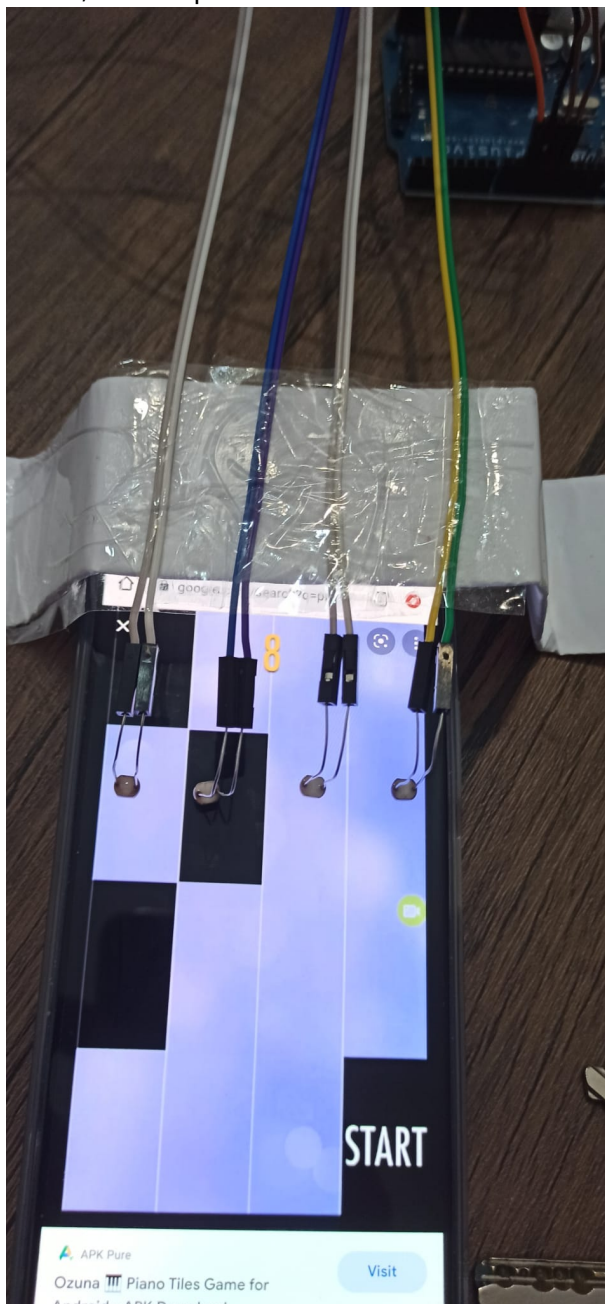
Pe ecran, monedele atasate de relay-urile 1 si 3 o sa fie mai departe de senzori, deci au un delay

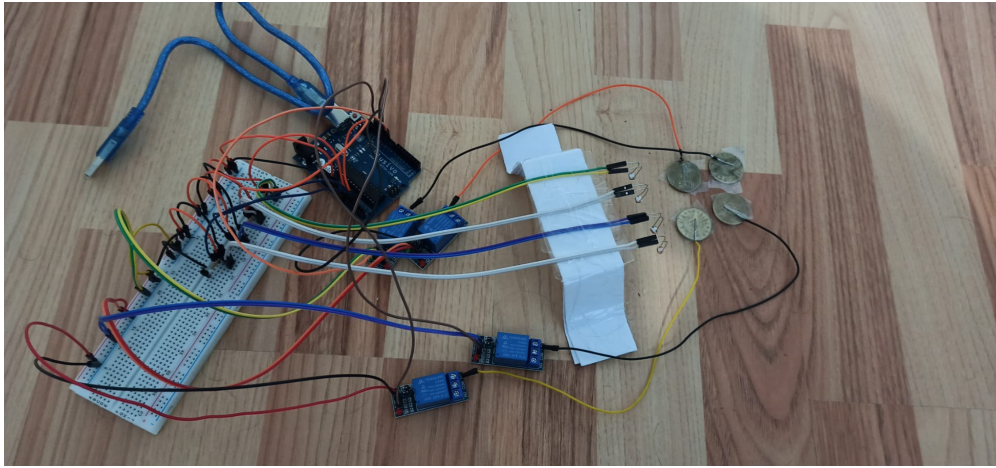
inainte de a face atingerea.

In cod trebuie adaugate delayuri custom pentru a lua in considerare distanta dintre LDR-uri (punctul de citire) si monedele de pe ecran (punctul de atingere al ecranului).

## Rezultate Obținute


In final, am reusit sa fac un robot care poate diferentia intre casutele negre si cele albastre de pe ecran, si care poate simula actiunea de 'touch' pe casutele negre.





## Concluzii

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

## Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

## Bibliografie/Resurse

[https://create.arduino.cc/projecthub/susheel\\_06/piano-tile-robot-a06ae6?ref=part&ref\\_id=8233&offset=5622](https://create.arduino.cc/projecthub/susheel_06/piano-tile-robot-a06ae6?ref=part&ref_id=8233&offset=5622)

<https://www.youtube.com/watch?v=8hIQ0MiowN8>

<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/avaduva/pianoplayer>



Last update: **2022/05/27 20:28**