

# Hand Tracking

Autor: Șerban Sorohan

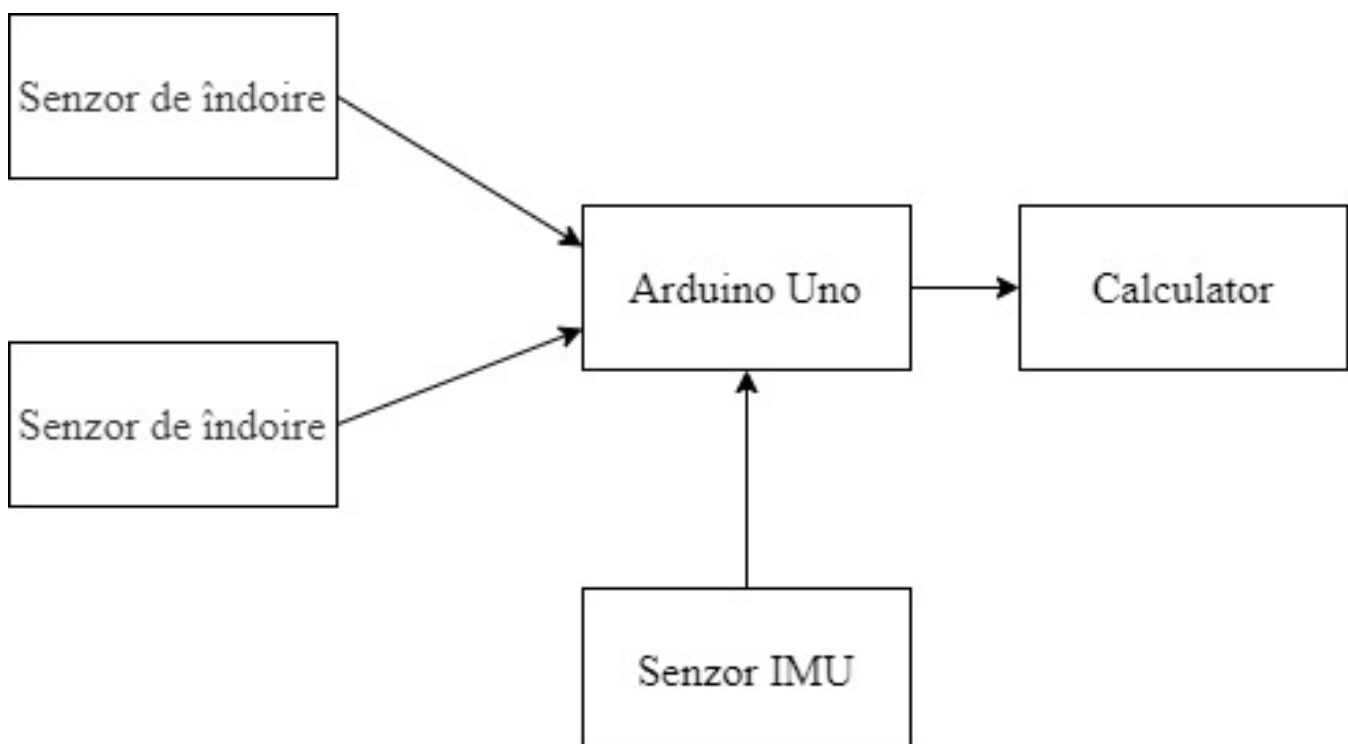
## Introducere

- O mână folosită pentru detecția mișcărilor și a gesturilor, de exemplu, detecția gesturilor pentru jocul piatră, hârtie, foarfecă.
- Scopul proiectului, pentru început este de a detecta mișcările necesare jocului piatră, hârtie, foarfecă.
- Consider că acest proiect poate fi extins și utilizat în mai multe locuri, precum: transformarea limbajului semnelor în text, control de la distanță a roboților care necesită îndemânare cât mai bună (ex. dezamorsare de bombe), folosirea mâinilor în aplicații de tipul virtual reality.

## Demo

[Demo](#)

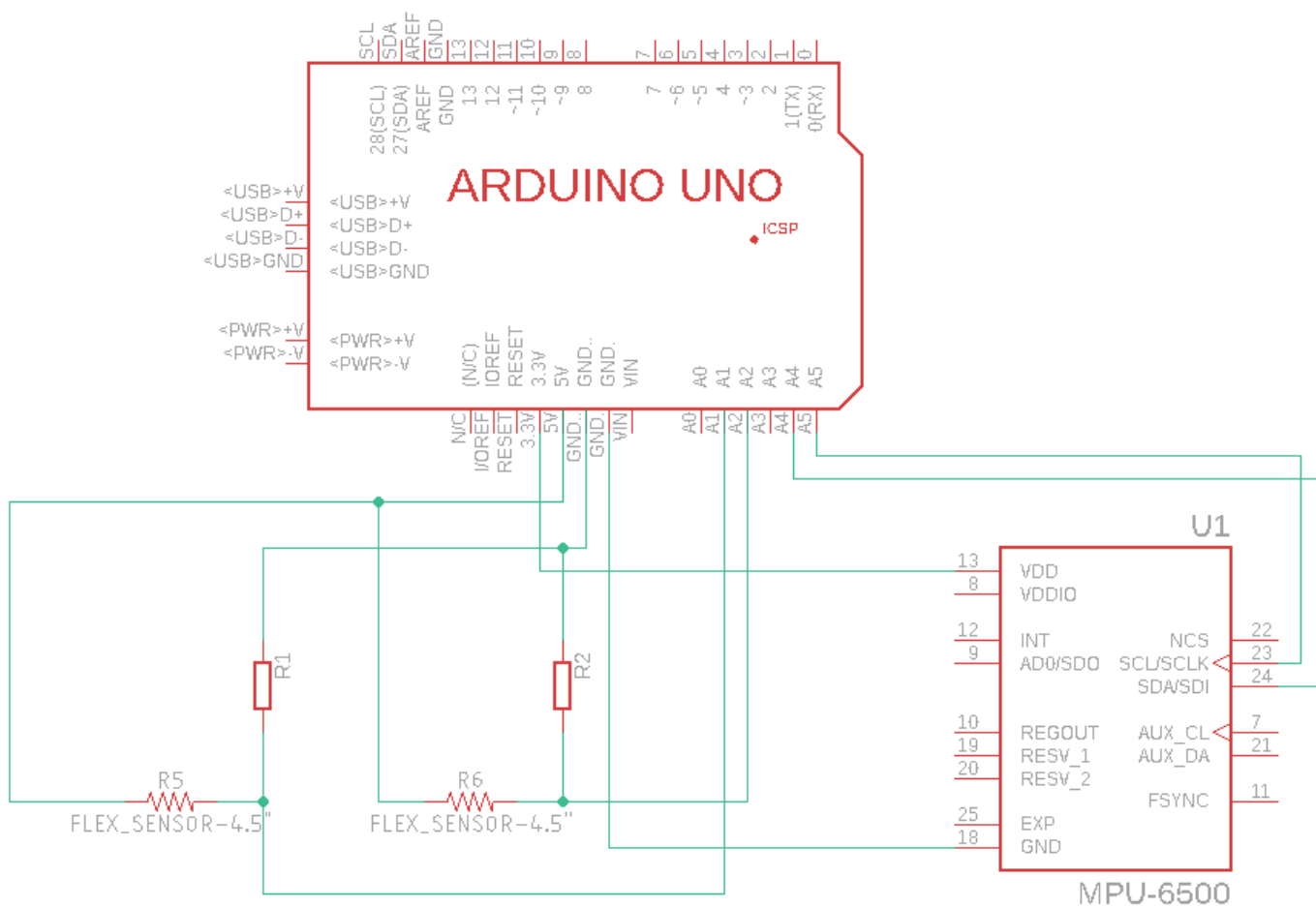
## Descriere Generala



- Senzorii de îndoire trimit către arduino gradul de îndoire al senzorului.
- Senzorul IMU trimite către arduino accelerațiile pe cele trei axe.
- Plăcuța arduino procesează datele, face o medie a valorilor trimise de către senzorii de îndoire, determină semnul făcut de către utilizator (piatră/hârtie/foarfecă) și alege un semn în mod aleator pentru plăcuță, apoi trimite către calculator un feedback vizual.

## Hardware Design

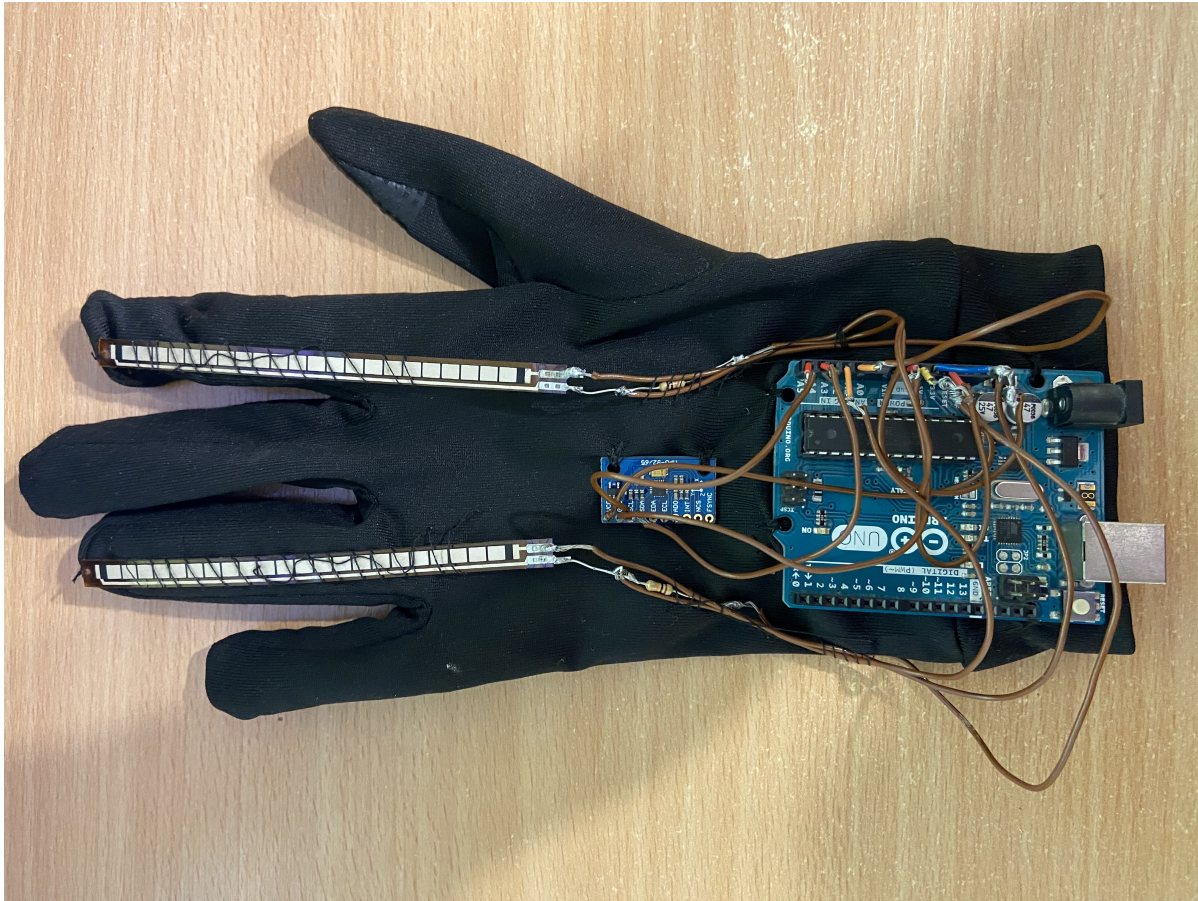
### Schemă electrică:



### Piese:

1. Arduino Uno
2. Senzor IMU
3. 2 x senzor de îndoire
4. Rezistențe
5. Mănușă

### Poză mănușă:



## Software Design

\* Codul este dezvoltat în IDE-ul oferit de către arduino și nu reprezintă o logică foarte complicată.

\* Codul folosit este disponibil în secțiunea "Download".

### Abordare generală:

1. Am definit o mașină de stări finită (prin variabila "State") care determină când se începe monitorizarea mișcărilor făcute de utilizator.
2. Când începe monitorizarea, se înregistrează "V\_SIZE" valori și se face media acestora pentru a determina ce semn a făcut utilizatorul. După ce se calculează media, se generează o valoare aleatoare care reprezintă semnul ales de arduino, se compară cele două semne și se determină cine este câștigătorul sau dacă este remiză.

### Funcția setup():

1. Se inițializează rata de transfer de date la 9600 baud.
2. Se inițializează și calibrează senzorul MPU6500.
3. Se setează pinii pentru senzorii flexori.

### Funcția loop():

1. Se efectuează calcule pentru a determina gradul de îndoire al fiecărui senzor flexor.
2. Se normalizează gradul de îndoire între intervalul [0, scale].

3. Se preiau datele de la senzorul MPU6500.
4. Dacă mașină de stări finită este în starea de monitorizare, se înregistrează cele "V\_SIZE" valori, câte una pe fiecare iterație a funcției loop(), iar la sfârșitul înregistrărilor se calculează media și se determină câștigătorul.
5. Se actualizează mașina de stări dacă este cazul.
6. Dacă timp de 2 secunde, mașina de stări nu a suferit nici o modificare, aceasta revine la starea inițială.

## Rezultate Obținute

Am obținut rezultate satisfăcătoare atât pe partea de hardware cât și pe partea de software. Mănușa are un aspect plăcut și detectează cu o acuratețe ridicată semnele făcute de utilizator astfel încât acesta se poate juca piatră, hârtie, foarfecă contra plăcuței fără probleme.

## Concluzii

A fost un proiect interesant din care simt că am învățat destul de multe lucruri. La început mă așteptam ca partea de hard să fie ceva mai ușoară decât partea de soft, însă s-a dovedit fix invers.

## Download

[Cod sursă](#)

[Code sursă pentru reprezentarea grafică](#)

## Jurnal

**[14/05/2021]** - Am început proiectul prin a înțelege cum funcționează senzorul MPU6500 cu ajutorul unui breadboard. Am scris un cod minimal pentru a obține valorile senzorului folosind biblioteca 'MPU6050\_tockn'.

**[15/05/2021]** - A doua zi am continuat prin a înțelege cum funcționează senzorii de flexiune. Tot folosind breadboard-ul am creat un divizor de tensiune și cu ajutorul unor bucăți de cod furnizate de sparkfun (vezi bibliografie) am testat funcționalitatea senzorilor. După testarea senzorilor am început să mă joc puțin cu multimetru, să măsoz rezistențele senzorilor, a rezistențelor și voltajul plăcuței.

**[16/05/2021 - 20/05/2021]** - Pauză de la proiect, alte teme necesitau atenția :(

**[21/05/2021]** - Încep să fac planul pentru poziționarea pe mănușă, fac câteva cusături de probă să văd cum ar sta mai bine componentele.

**[22/05/2021]** - Încep să mă joc cu pistolul de lipit. Mă chinui câteva ore bune (nu sunt prea îndemânic) să lipesc la fiecare senzor de flexiune câte o rezistență și trei fire pentru a crea divizorul de tensiune.

**[23/05/2021]** - Cos pe mânușă senzorii de flexiune și plăcuța și fac câteva teste pentru a vedea dacă merg bine senzorii.

**[24/05/2021 - 25/05/2021]** - Pauză de relaxare, am muncit destul :).

**[26/05/2021]** - Încep să fac lipituri pentru senzorul MPU6500, lucru ce îmi ia destul de mult timp fiindcă sunt necesare lipituri destul de mici (între timp, pistolul de lipit începe să scoată fum :( ).

**[27/05/2021]** - Termin lipiturile pentru MPU6500 și refac cusăturile pentru cei doi senzori de flexiune, a.î. să fie poziționați mai bine pe mânușa. Fac câteva cusături și în jurul firelor pentru a fixa în anumite poziții care nu încurcă.

**[28/05/2021]** - Fac câte o gaură în vârful fiecărui senzor de flexiune (pe partea de plastic, nu în senzorul propriu-zis) pentru a îi fixa și mai bine pe mânușă și mă apuc să cos și senzorul MPU6500.

**[29/05/2021]** - Termin de cusut senzorul MPU6500 și încep să organizez partea de cod. Încep să folosesc biblioteca 'TinyMPU6050' în loc de 'MPU6050\_tockn' din motive de simplitate. Termin de scris mare parte din cod.

**[30/05/2021 - 01/06/2021]** - Pauză pentru a face tema la SO.

**[02/06/2021]** - Ordenez codul și mai umblu puțin la un senzor flexor care nu face bine o conexiune, de asemenea încep și scrierea jurnalizării (the journal is self aware!!!).

**[03/06/2021]** - Am refăcut legăturile firelor cu plăcuța deoarece câteodată erau întreruperi de semnale.

**[04/06/2021]** - Am făcut demo-ul și am finalizat pagina de wiki.

## Bibliografie/Resurse

- [https://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/hand\\_tracking?do=export\\_pdf](https://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/hand_tracking?do=export_pdf)
- <https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide>

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

Permanent link:  
[http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/hand\\_tracking](http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/hand_tracking)



Last update: **2021/06/04 09:20**