

**Luchian Radu**

· Mail: [luchian\\_radu98@yahoo.com](mailto:luchian_radu98@yahoo.com)



**Adomniți Robert**

· Mail: [robert-alexandru.adomniti@student.tuiasi.ro](mailto:robert-alexandru.adomniti@student.tuiasi.ro)



**Macovei Dănuț**

· Mail: [danutmcv@yahoo.com](mailto:danutmcv@yahoo.com)



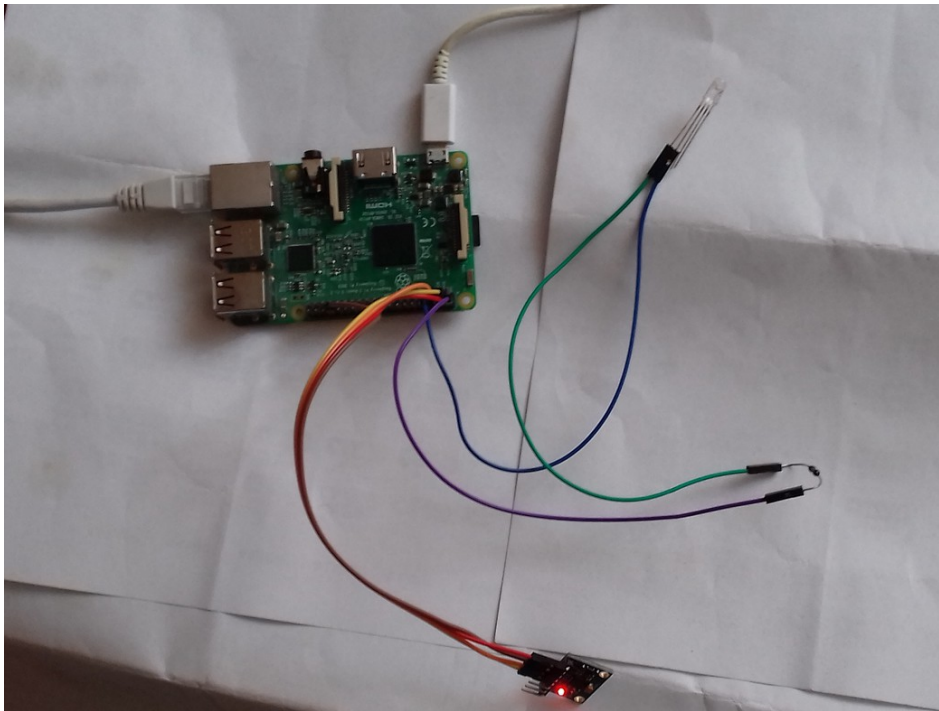
**Cucutianu Robert**



· Mail: [rcucutianu@gmail.com](mailto:rcucutianu@gmail.com)

**Seismograf web cu alarmă și lumini de avertizare**

Poză:



### **Elevator pitch:**

Acest proiect are ca scop realizarea unui seismograf care poate fi urmărit pe web cu ajutorul unui browser. Acesta va emite semnale luminoase de alarma in cazul depistarii unui cutremur. Acest proiect contine programare web front end (javascript) și back-end cu Python pe Raspberry Pi.

### **Componente hardware:**

- Raspberry Pi 3 Model B
- Accelerometru MPU6050
- Led rosu
- Cabluri
- Rezistor

## Componente software:

- Putty
- Python3
- Flask & smlib pentru Python3
- Browser web

## Poveste:

Acest proiect are ca scop realizarea unui seismograf care poate fi urmărit pe web cu ajutorul unui browser. Acesta va emite semnale luminoase de alarma in cazul depistarii unui cutremur. Pentru a realiza acest lucru proiectul funcționează în modul următor: in python, cu biblioteca smlib se preiau date de pe accelerometrul MPU6050. De asemenea, tot în python este făcut un server web în flask care servește cererile de la browser. Acesta are o rută care va returna valoarea curentă citită de pe accelerometru.

La client, în browser se va afișa un grafic făcut cu biblioteca Chart.js. Aceasta a fost aleasă pentru simplitatea de folosire și pentru aspectul plăcut. Codul javascript va face câte un request, va desena valoarea și va aștepta 200ms.

Pentru serverul web am optat pentru Flask în defavoarea altor alternative cum ar fi Apache pentru simplitatea de utilizat și pentru ca codul necesar pentru realizarea acestui scop este scurt și la obiect.

În faza de dezvoltarea conectarea la mașina Raspberry Pi s-a făcut utilizând Putty prin SSH.

De asemenea, serverul cand citeste o valoare peste limita admisă (THRESHOLD\_CUTREMUR) înseamnă ca este detectata o condiție de cutremur și va porni alarma vizuala cu ledul rosu care clipește. Pentru acest scop am creat un fir de execuție separat care când primește un semnal de alarma trimite impulsuri la led pentru ca acesta sa clipeasca o perioada de timp și după se oprește, asteptand alt semnal de

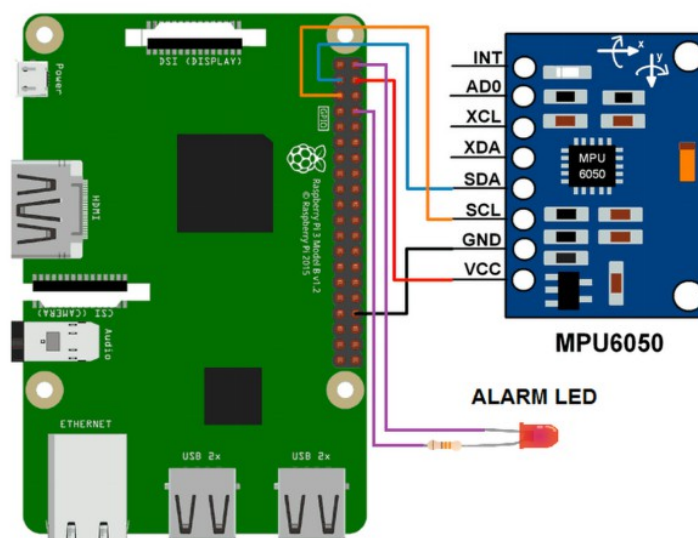
același tip.

**Proiectul poate fi observat în acțiune în următoarele videoclipuri:**

• **Proiectul fizic:** <https://streamable.com/e1o6a2>

• **Interfața grafică:** <https://streamable.com/m9fvys>

**Schematic:**



**Cod:** [https://github.com/RaduLRP/PROIECT\\_SM](https://github.com/RaduLRP/PROIECT_SM)

## Contribuția fiecăruia

Luchian Radu - partea hardware + schematică + configurare sistem de operare

Robert Adomniți - interfața web + declanșare alarmă

Dănuț Macovei - preluare date accelerometru

Robert Cucutianu - conexiunea între interfața web și date de pe accelerometru