

# Poligraf

## Introducere

Un poligraf este un aparat care masoara anumiti parametrii fiziologici ai unei persoane si poate detecta daca acea persoana minte sau spune adevarul la adresarea unei intrebari. Cu ajutorul unui senzor de puls si al unui de temperatura si umiditate, voi putea determina sinceritatea unui raspuns.

## Descriere generală

Poligraful pe care as vrea sa-l realizez, necesita ca o persoana sa fie conectata la senzorul de puls si cel de temperatura si umiditate in timp ce i se pune o intrebare. In functie de sinceritatea raspunsului, prin lcd, difuzor si led-uri voi semnaliza care este verdictul, daca persoana a mintit sau a fost sincera. Led-urile vor fi de 3 feluri, verde = sincer, rosu = minciuna, galben = incert. Pe LCD, voi scrie intrebarea adresata, cat si raspunsul si, de asemenea, pulsul persoanei. Difuzorul va semnaliza daca persoana a mintit sau nu in momentul afisarii raspunsului.



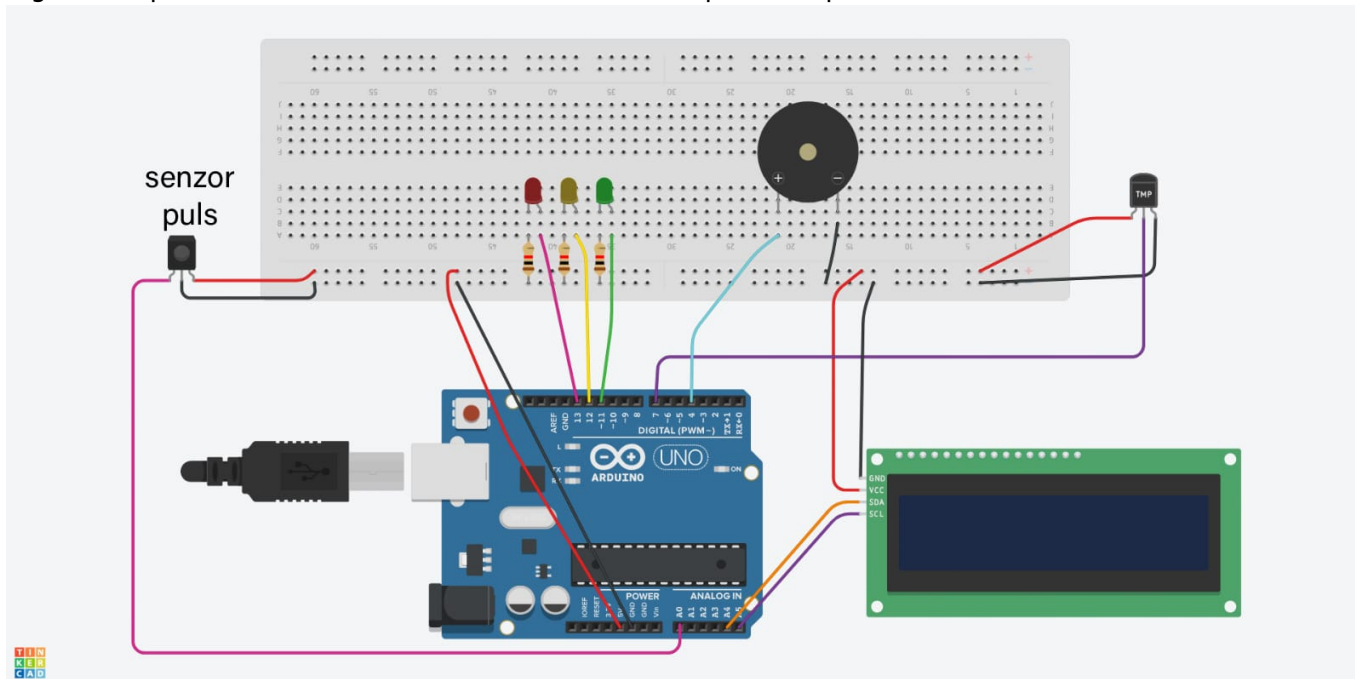
## Hardware Design

### Lista de piese:

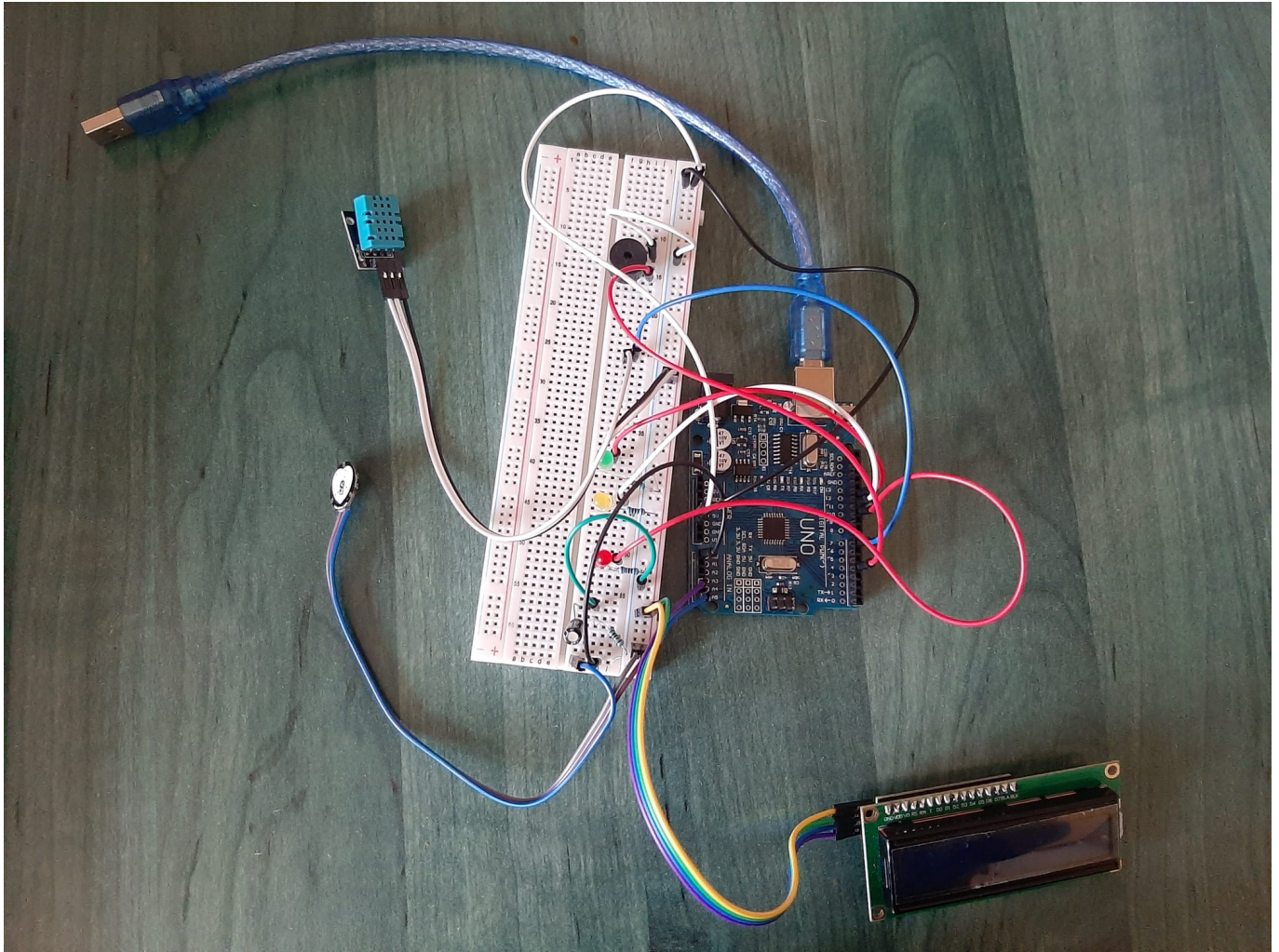
- Arduino UNO
- LCD 16x2(I2C)
- 3 LED-uri(rosu, verde, galben)
- Senzor puls
- Senzor temperatura si umiditate
- Breadboard
- Buzzer
- 3 rezistente 1K

Am folosit Arduino UNO pentru a gestiona toate componentele si a interpreta datele primite de la senzori. Pe LCD voi afisa intrebarea, rezultatul daca este sincer sau nu pe prima linie si pe a doua, pulsul. Acesta are 4 pini, VCC, GND, SDA conectat la A4 si SCL conectat la A5. Cu ajutorul senzorului de puls, voi prelua informatia despre bataile inimii si impreuna cu senzorul de temperatura si umiditate voi putea interpreta rezultatele. Senzorul de umiditate are 3 pini, VCC, GND si OUT care e

conectat la pin 7, aleas aleator. Cel de puls are, de asemenea, 3 pini VCC, GND si OUT care e conectat la A0. Am folosit cele 3 led-uri pentru a arata care este verdictul, care sunt conectate la 3 pini, impreuna cu 3 rezistente de 1K. Buzzerul este legat la un pin si la gnd si il folosesc la fel pentru a semnaliza verdictul. Pinii alesi pentru led-uri si difuzor sunt alesi aleator din intervalul 2-13 dintre cei digitali si apoi in cod le-am asociat eu insemnatatea pentru a produce rezultatul dorit.



Pentru a putea avea o claritate mai buna a semnalului si a reduce zgomotul pentru senzorul de puls, am adaugat si un filtru trece-jos. Montajul final este acesta:



## Software Design

Am folosit Visual Studio Code ca mediu de dezvoltare.

Input user: Am implementat 2 modalitati de primire a intrebarilor, ori de la seriala folosind `Serial.read()`, ori am un set de intrebari prestabilite.

Biblioteci:

- DHT sensor library
- PulseSensor Playground

Functii:

- `setup`
- `loop`
- `method1`(implementare pentru set prestabilit)
- `method2`(intrebare de la seriala)
- `get_params`(care inregistreaza valorile senzorilor)

In setup: am initializat LCD, senzorul de temperatura si umiditate, pinii pentru led-uri i-am setat ca output si senzorul de puls. La inceput am o perioada de 50000 in care voi inregistra pulsul din deget si umiditatea palmei si voi calcula o valoare medie pe care o voi compara apoi cu valoarea inregistrata dupa adresarea problemei.

In get\_params: am calculat media pe o perioada de 30000 inregistrez valori si calculez apoi o medie a lor

In method1/method2: voi compara valoarea initiala cu cea de dupa punerea intrebarii, folosind o marja de eroare.

In loop: apelez metodele in functie de modul specificat de "pacient" si apoi calculez niste statistici pentru cat de onest a fost.

[Demo](#)

## Rezultate Obținute si Concluzii

Proiectul pentru a functiona 100% corect ar avea nevoie de niste senzori cu o precizie mare, iar senzorii pe care ii am eu nu ofera neaparat acuratetea dorita. Am incercat sa ii fac sa aiba o precizie mai mare din cod pentru a putea da un verdict, ceea ce a adus o imbunatire semnificabila. Poate da astfel rezultatele asteptate, desi nu o face mereu.

## Download

Arhiva: [mincaecaterina\\_proiectpm.zip](#)

## Jurnal

- 10.04 - alegere tema proiect
- 10.05 - asamblare tuturor componentelor
- 17.05 - adaugare filtru trece jos pentru reducerea zgomotului
- 22.05 - terminare implementare software

# Bibliografie/Resurse

## Resurse

- <https://docs.arduino.cc/learn/>
- <https://how2electronics.com/pulse-rate-bpm-monitor-arduino-pulse-sensor/>
- [https://projecthub.arduino.cc/arduino\\_uno\\_guy/i2c-liquid-crystal-displays-5eb615](https://projecthub.arduino.cc/arduino_uno_guy/i2c-liquid-crystal-displays-5eb615)
- <https://www.ardumotive.com/how-to-use-a-buzzer-en.html>
- <https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/>
- [https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/?utm\\_content=cmp-true#google\\_vignette](https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/?utm_content=cmp-true#google_vignette)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/rpascale/ecaterina.minca>



Last update: **2024/05/28 09:24**