

PROIECT ACHIZITII DE DATE

SENZOR DE PROXIMITATE

Link simulare TinkerCad: /*TODO*/

STUDENT: PETRUC RARES

GRUPA: 322CD

Introducere

În contextul pandemic actual, am luat o serie de măsuri foarte importante, printre care putem enumera și distanțarea fizică. La orice moment din zi, știm că trebuie să ținem o distanță între 1.5m-2m de alte persoane, însă nu mereu o și facem. Drept urmare, am creat o aplicație ce te avertizează în cazul în care te apropii prea mult de persoanele din jur, oferindu-ți și informații despre distanța minimă la care te afli acum față de alte persoane, după criteriile pe care le vei prezenta mai jos.

Evident, obiectivul este de a mări distanța, atunci când aceasta nu este tocmai respectată, pentru a combate răspândirea virusului COVID-19.

SCHEMA CIRCUITULUI

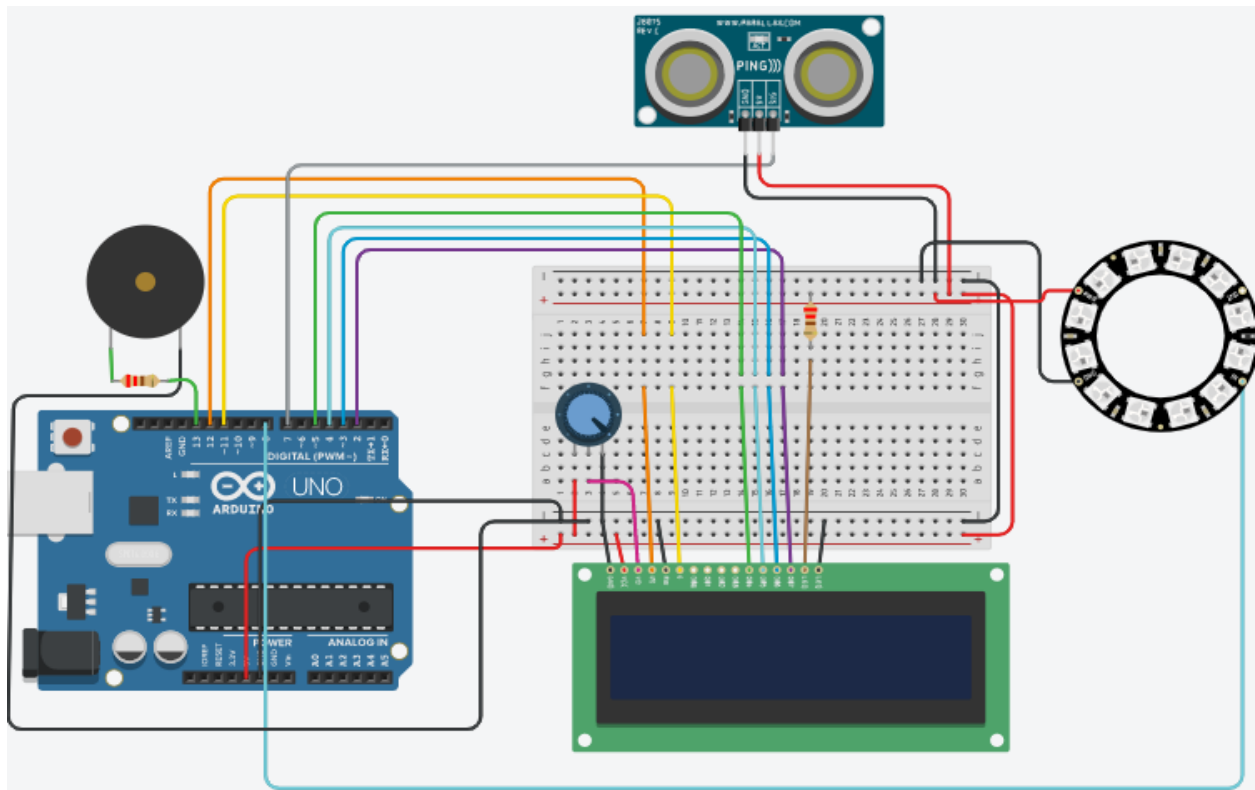


Fig. 1 Schema circuitului de tip „Breadboard View”.

Deși dependente în cod pentru scopul aplicației, în ceea ce privește strict legarea în circuit, componentele principale nu sunt interconectate, așa cum se vede și în figura 1. Practic, fiecare componentă este legată la arduino, respectiv Breadboard, iar singurele locuri în care se intersectează sunt potențialele de ground și power (unde este cazul). Am folosit documentația oficială pentru fiecare dintre componentele folosite cu Arduino: LCD [3], BUZZER cu funcția aferentă de tone [4], senzorul ultrasonic de distanță [1], etc.

APLICATIA

În bucla loop din codul pentru Arduino, am început prin a citi datele despre distanță prin apelul funcției `getDuration`. Am înțeles și folosit codul din documentația oficială pentru arduino a senzorului ultrasonic de distanță din Tinkercad([1, 2])). Ulterior, am convertit rezultatul returnat de funcție în centimetri, respectiv inci, pentru a putea afișa rezultatul pe ecranul LCD într-un mod cât mai sugestiv, funcție de zona în care va fi folosit produsul.

Pentru simplitate în dezvoltare, m-am folosit de reprezentarea cu care eram eu familiar, în centimetri, pentru a decide în care caz de siguranță se afla utilizatorul. Astfel, am stabilit următoarele praguri de siguranță: VERDE pe NeoPixel Ring, respectiv mesajul „SAFE” pe LCD, în caz de distanță minimă față de persoanele din jur este de peste 1.5 m.

GALBEN pe NeoPixelRing, respectiv mesajul „RAISE DISTANCE” pe LCD, în caz de distanță minimă față de persoanele din jur este între 1 m și 1.5 m.

ROSU pe NeoPixelRing, respectiv mesajul „DANGER!” pe LCD, în cazul în care distanță minimă față de persoanele din jur este de sub 1 m.

Mai mult decât atât, în apropierea și mai bună a distanței, am ales să aprind pentru fiecare caz în parte (VERDE, GALBEN și ROSU), un număr de leduri pe NeoPixelRing ce anunță apropierea de următorul prag sau nu, astfel:

Pentru VERDE:

- pentru distantele mai mari de 2 m, vor fi aprinse 4 leduri.
- pentru distantele intre 1.75 si 2 m, vor fi aprinse 8 leduri.
- pentru distantele intre 1.5 si 1.75 m, vor fi aprinse toate cele 12 leduri, anuntand astfel apropierea de o posibila trecere la urmatorul prag de siguranta.

Similar, pentru GALBEN:

- pentru distantele cuprinse intre 1.33 si 1.50 m, vor fi aprinse 4 leduri;
- pentru distantele cuprinse intre 1.16 si 1.33 m, vor fi aprinse 8 leduri;
- pentru distantele cuprinse intre 1 si 1.16 m, vor fi aprinse cele 12 leduri.

, pentru ROSU:

- pentru distantele cuprinse intre 0.75 si 1 m, vor fi aprinse 4 leduri;
- pentru distantele cuprinse intre 0.5 si 0.75 m, vor fi aprinse 8 leduri;
- pentru distantele de sub 0.5 m, vor fi aprinse cele 12 leduri;

In plus, pentru cazul ROSU, de pericol, indiferent de distanta la care ne aflam, se va activa si Buzzer-ul cu rol in captarea atentiei si trezirea interesului, numai ca si in alt mod decat vizual, si anume auditiv.

Am fost ajutat de componentele folosite, in ceea ce priveste atat senzorii, si aici ma refer la cel ultrasonic, folosit pentru distanta, intrucat range-ul sau variaza intre 2 cm si 3 m, ideal pentru o astfel de aplicatie. Iar, la Buzzer, am reusit sa il fac sa nu intre in conflict cu afisarea mesajelor de pe led. Asa, cum scrie si in documentatia sa oficiala [4], functia tone va afecta pinii 3 si 11 ai placii folosite in simulare, acest lucru determinand functionarea defectuoasa a lcd-ului, generand mesaje cu precizie mai slaba. Asadar, am folosit direct constructia digitalWrite(BUZZERPIN, HIGH), pentru a nu avea interferente nedorite.

Atasez in Fig. 2 un exemplu de rulare a circuitului pentru cazul ROSU, in care se afiseaza mesajul „DANGER!” pe LCD si in care avem cele 8 leduri rosii, intrucat distanta este cuprinsa intre 0.5 si 0.75m, iar Buzzer-ul suna:

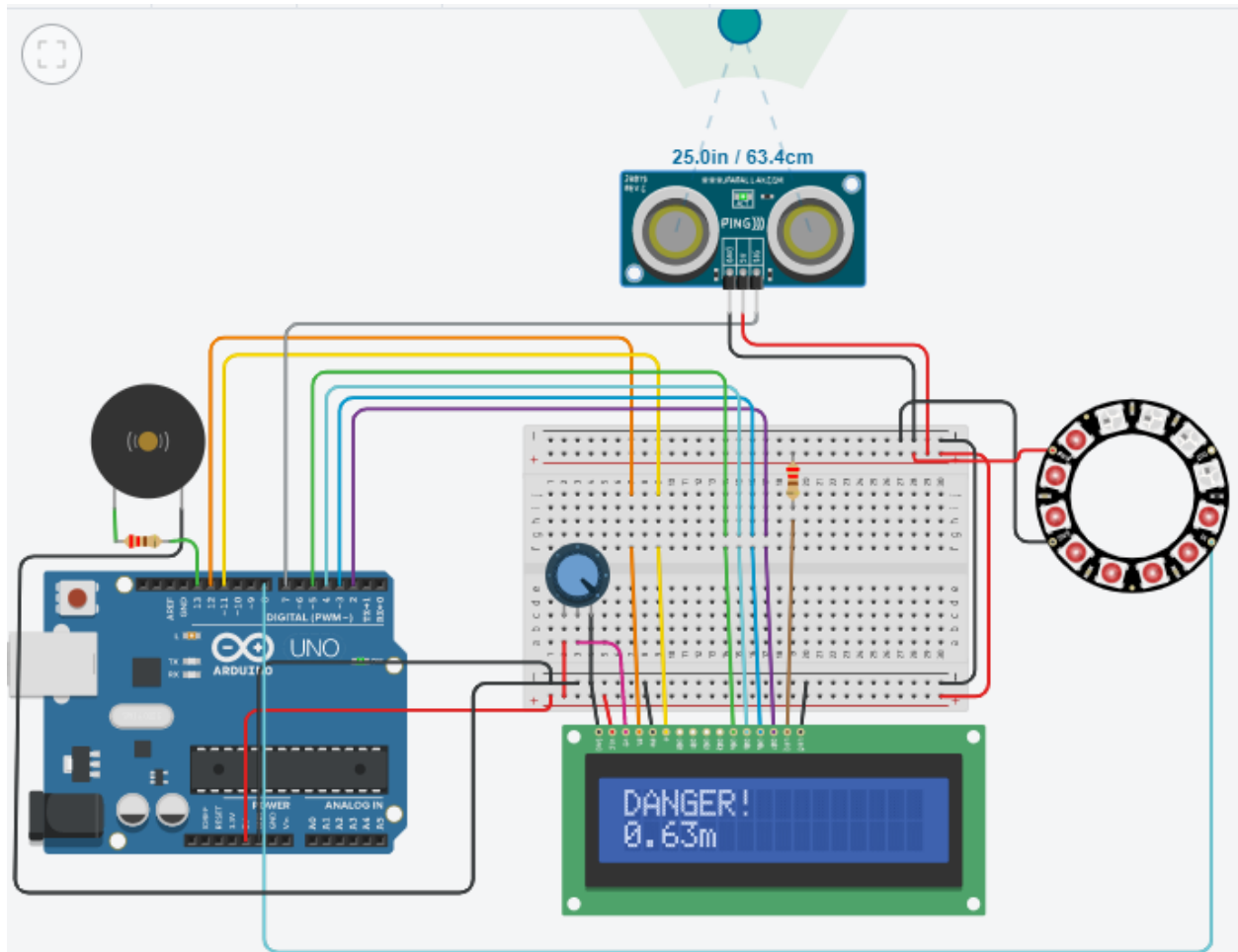


Fig. 2: Exemplu de rulare a aplicatiei pentru o distanta de 0.63m.

Referinte:

- [1]: <https://www.arduino.cc/en/Tutorial/BuiltInExamples/Ping>.
- [2]: <https://www.cypress.com/file/55476/download>.
- [3]: <https://www.arduino.cc/en/Tutorial/LibraryExamples/HelloWorld>.
- [4]: <https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>.