Alarma de proximitate folosind

un senzor de tip PIR

Naumencu Mihai – 322CD

1. **Introducere**

Alarma de proximitate este implemetata folosind un senzor de tip PIR (passive infrared sensor). Acesta detecteaza miscarea intr-o anumita raza si, in cazul detectarii, un buzzer de tip piezo se va activa si va ramane activat atat timp cat miscarea continua. In timpul in care buzzerul este activ, durata soneriei va fi afisata pe un ecran LCD 16 x 2. Cat timp senzorul nu detecteaza miscare, alarma este in starea de stand by si acest fapt va fi afisat pe ecranul LCD. Alarma poate fi activata sau dezactivata prin intermediul unui switch ON/OFF.

1. **Schema circuitului**

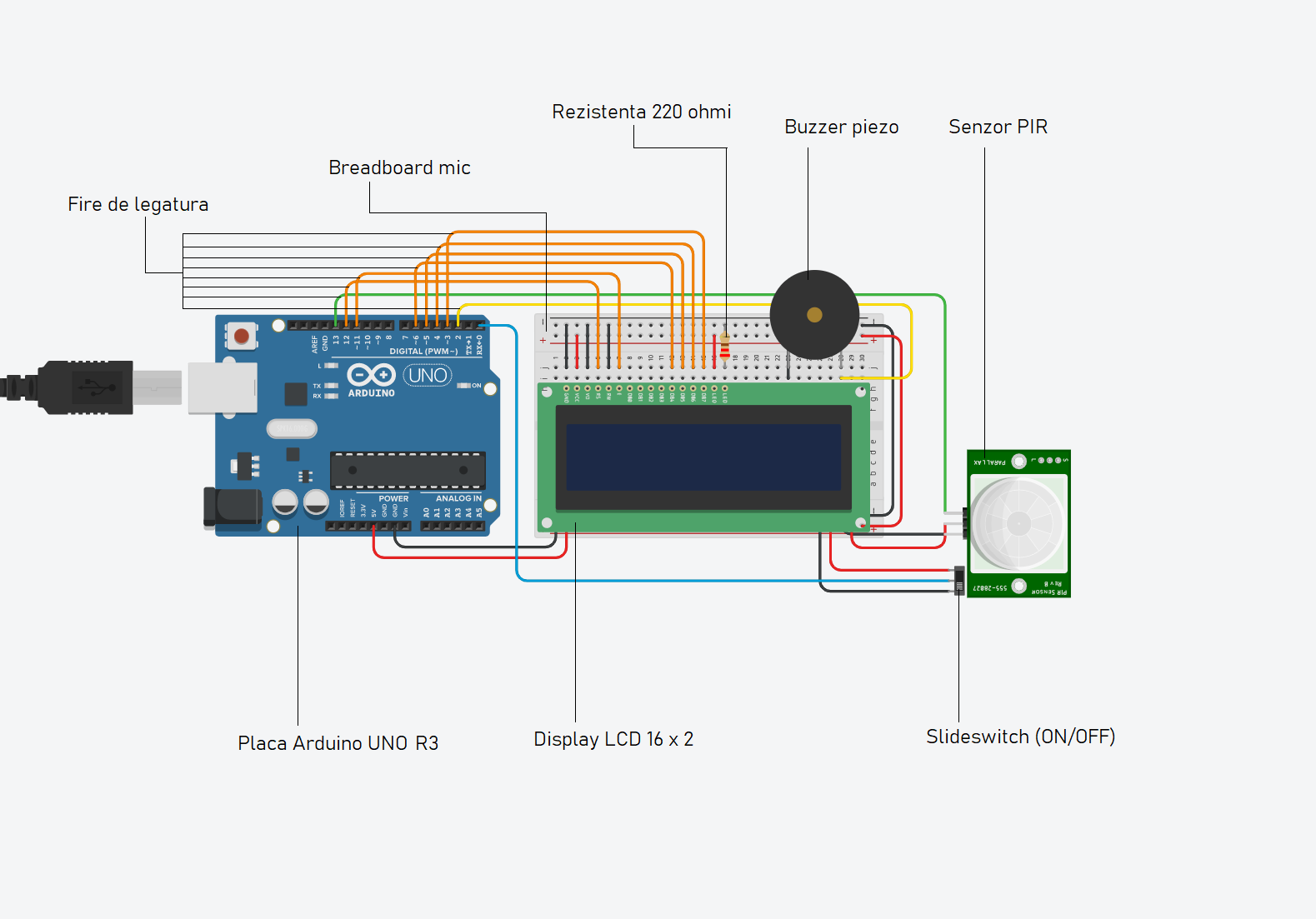


Diagrama circuitului (realizat in tinkerCAD)

Alarma contine urmatoarele componente de circuit: o placa Arduino UNO R3, un breadboard, un sezor de tip PIR, un buzzer piezo, un ecran LCD 16 x 2, un slideswitch ON/OFF, o rezistenta de 220 ohmi si firele prin intermediul carora se realizeaza legaturile dintre aceste componente.

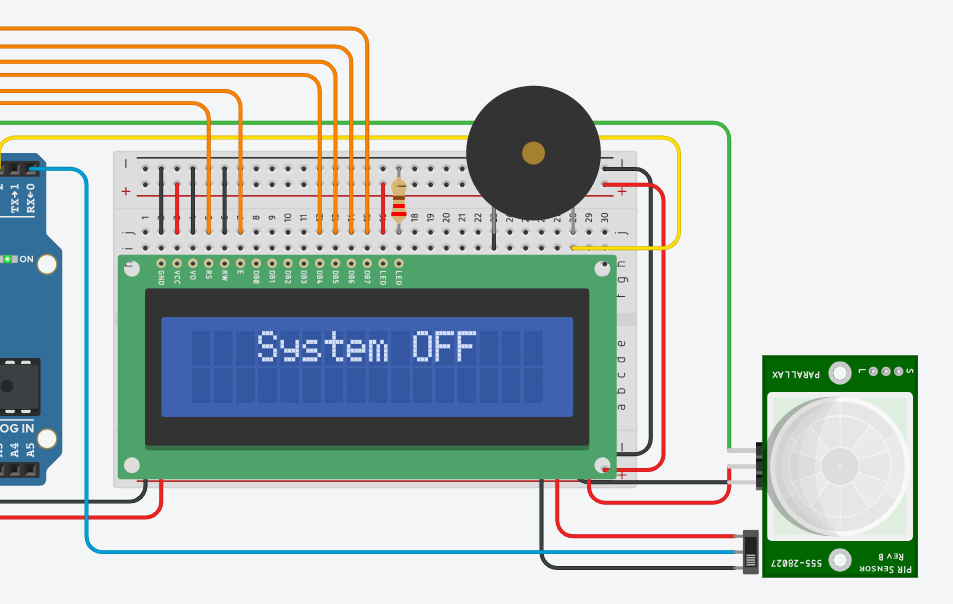
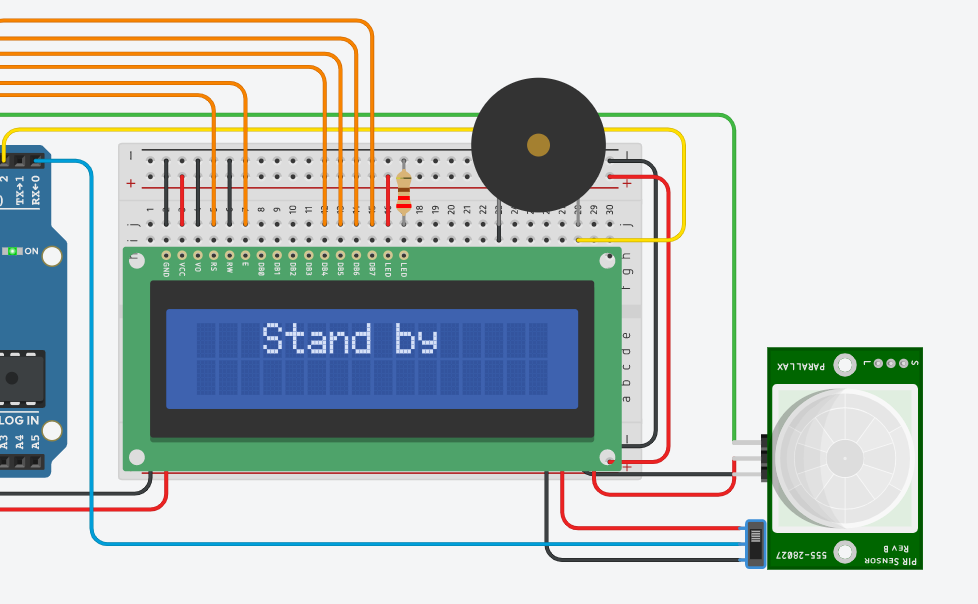
Legarea componentelor:

* Breadboard:
* Bus-ul + este conectat la alimentarea de 5V a placii Arduino
* Bus-ul – este conectat la un pin de ground al placii Arduino (GND)
* Buzzer:
* Borna *pozitiva* este legata la -
* Borna *negativa* este legata la pin-ul 2 al placii Arduino (pin de output)
* Senzor PIR:
* Borna *ground* este legata la -
* Borna *power* este legata la +
* Borna *sognal* este legata la pin-ul 13 al placii Arduino (pin de input)
* Slideswitch ON/OFF:
* Borna *terminal 2* este legata la -
* Borna *terminal 1* este legata la +
* Borna *common* este legata la pin-ul 0 al placii Arduino
* Ecran LCD:
* Pozitionat direct pe breadboard
* GND la -
* VCC la +
* V0 la -
* RS la pin 12
* RW la -
* E la pin 11
* DB4 la pin 6
* DB5 la pin 5
* DB6 la pin 4
* DB7 la pin 3
* LED anode la +
* LED cathode – (prin rezistenta de 220 ohmi)

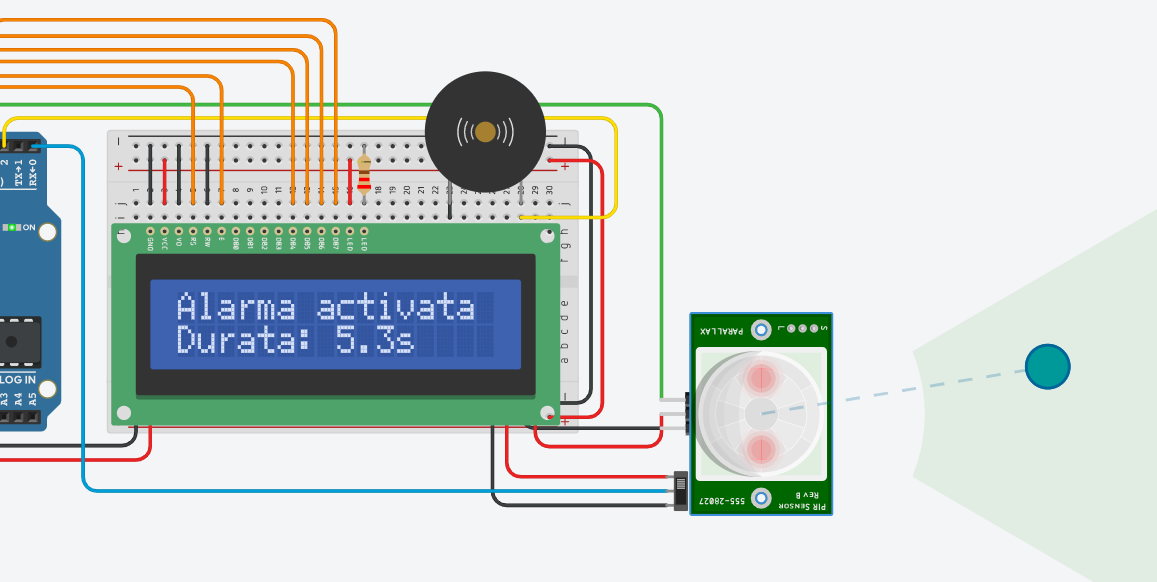
Color coding-ul firelor de logatura:

* Firele negre fac legatura de la un pin de ground al placii Arduino (GND) si bus-ul "-" al breadboardului, cat si legaturile componentelor cu acesta
* Firele rosii fac legatura de la pin-ul de alimentare de 5V al placii Arduino si bus-ul "+" al breadboardului, cat si legaturile componentelor cu acesta
* Firele portocalii fac legaturile dintre ecranul LCD si pinii de control aflati pe placa Arduino
* Firul albastru face legatura intre borna *common* a slideswitch-ului si pin-ul 0 al placii Arduino
* Firul galben face legatura intre borna *negative* a buzzer-ului piezo si pin-ul 2 al placii Arduino (pin de output)
* Firul verde face legatura intre borna *signal* a senzorului PIR si pin-ul 13 al placii Arduino (pin de input)

1. **Aplicatia**



Slideswitch-ul se afla pe pozitia OFF Slideswitch-ul se afla pe pozitia ON, sistemul este in starea de stand by



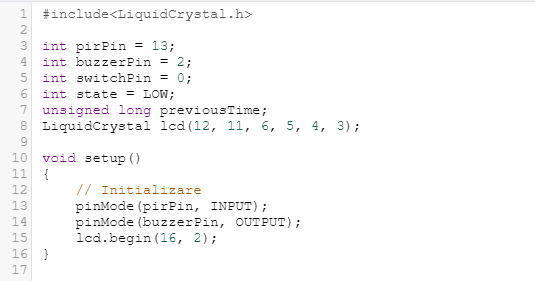
Senzorul detecteaza miscare in raza sa de actiune

Sistemul de alarma se poate afla in 3 stari (prezentate in figurile de mai sus): slideswitch-ul este pe pozitia OFF, asadar sistemul este oprit (ecranul LCD afiseaza mesajul "System OFF"), slideswitch-ul este pe pozitia ON si senzorul nu a detectat inca miscare in raza sa de actiune (ecranul LCD afiseaza mesajul "Stand by") sau slideswitch-ul este pe pozitia ON si senzorul detecteaza miscare in raza sa de actiune (ecranul LCD va afisa mesajul "Alarma activata", urmat de durata actualei detectari).

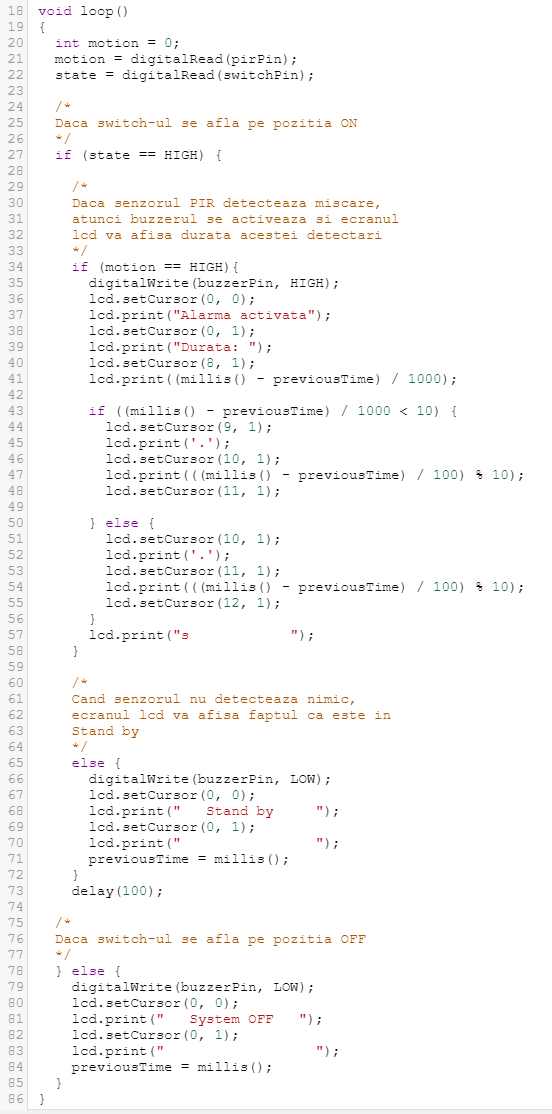
Descrierea componentelor:

* Placa Arduino UNO R3 – o placa microcontroller construita pe o platforma open-source dotata cu 14 pini I/O digitali (6 capabili de output PWM) si 6 pini I/O analogi.
* Ecran LCD 16 x 2 – folosit pentru afisaj electronic, poate afisa caracatere pe 2 randuri a cate 16 spatii
* Slideswitch – un switch simplu folosit pentru pornirea/oprirea sistemului de alarma
* Buzzer piezo – sonerie piezoelectrica, aceasta este activata la in cazul detectarii miscarii de catre senzorul PIR
* Senzor PIR – senzor electronic ce masoara radiatia luminii infrarosii din raza sa de actiune, folosit pentru detectarea miscarii

Implementarea software:



In partea de setup a aplicatiei sunt initializati pinii diverselor componente (pin-ul senzorului PIR va fi setat ca pin de input, in timp ce pin-ul buzzer-ului va fi setat ca pin de output), variabile (*state* pentru determinarea starii switch-ului si *previousTime* pentru resetarea contorului de durata la fiecare activare) cat si ecranul LCD (conform librariei *LiquidCrystal.h*).

In corpul *loop:*

In variabilele *motion* si *state* se vor stoca starile pinilor aferenti senzorului PIR si switch-ului.

In cazul in care switch-ul se afla pe pozitia *ON,* se va verifica starea senzorului PIR, daca aceasta este *HIGH* (senzorul a detectat miscare in raza sa de actiune), se va trimite semnalul catre buzzer pentru ca acesta sa se activeze si se va afisa pe ecranul LCD mesajul "Alarma activata", urmat de durata actualei activari. Daca senzorul nu detecteaza miscare se va afisa mesajul "Stand by".

In cazul in care switch-ul se afla pe pozitia *OFF*, se afiseaza pe ecranul LCD mesajul "System OFF".

Se stocheaza in variabila *previousTime* timpul curent pentru afisare corecta a duratei in cazul in care, in urmatorul ciclu, alarma este activata.

Referinte:

<https://www.circuitbasics.com/how-to-set-up-an-lcd-display-on-an-arduino/>

<https://create.arduino.cc/projecthub/electropeak/pir-motion-sensor-how-to-use-pirs-w-arduino-raspberry-pi-18d7fa>