

Home Security System

Autor:

Cocoru Emilia-Iuliana

Introducere

Sistemul de alarmă vine în ajutorul celor cărora le este frică să stea singuri în casă, simțindu-se amenințați de apariția unor persoane străine sau de alți factori externi ce le-ar putea pune viața în pericol.

Scopul proiectului constă în detectarea scurgerilor de gaz, cât și supravegherea perimetrului din fața ușii, oferind un grad de protecție ridicat.

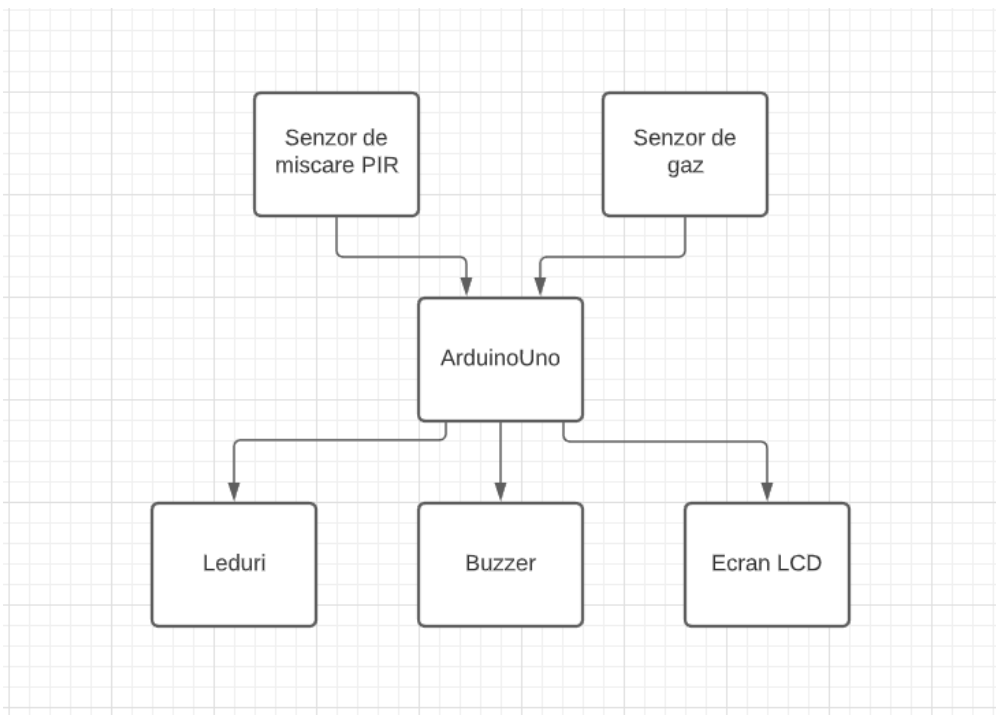
Fiind o persoană paranoică, ideea mi-a venit din dorința de a mă simți în siguranță atunci când mă aflu singură în casă, în special pe timpul nopții.

Proiectul mi se pare util atât pentru a-mi pune în practică noțiunile învățate la laborator, cât și pentru rolul acestuia în viața de zi cu zi. Este extrem de important să avem grijă de siguranța noastră, mai ales când tehnologia ne oferă atât de multe posibilități de a ne proteja.

Descriere generală


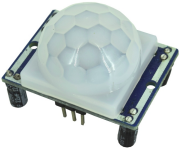
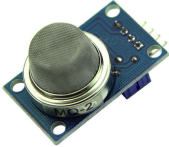

Sistemul meu de alarma este alcătuit dintr-un senzor de mișcare PIR și un senzor de gaz care comunică cu placa ArduinoUNO. O dată ce acestea detectează mișcare/gaz vor activa un buzzer și semnalele luminoase ale unor leduri în așa fel încât proprietarul să fie avertizat de pericol. De asemenea, pe ecranul LCD va apărea mesajul cu pericolul (dacă a fost activat senzorul de mișcare/gaz).


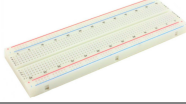
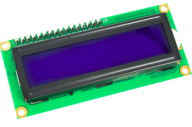

Diagrama bloc corespunzătoare proiectului:



Hardware Design

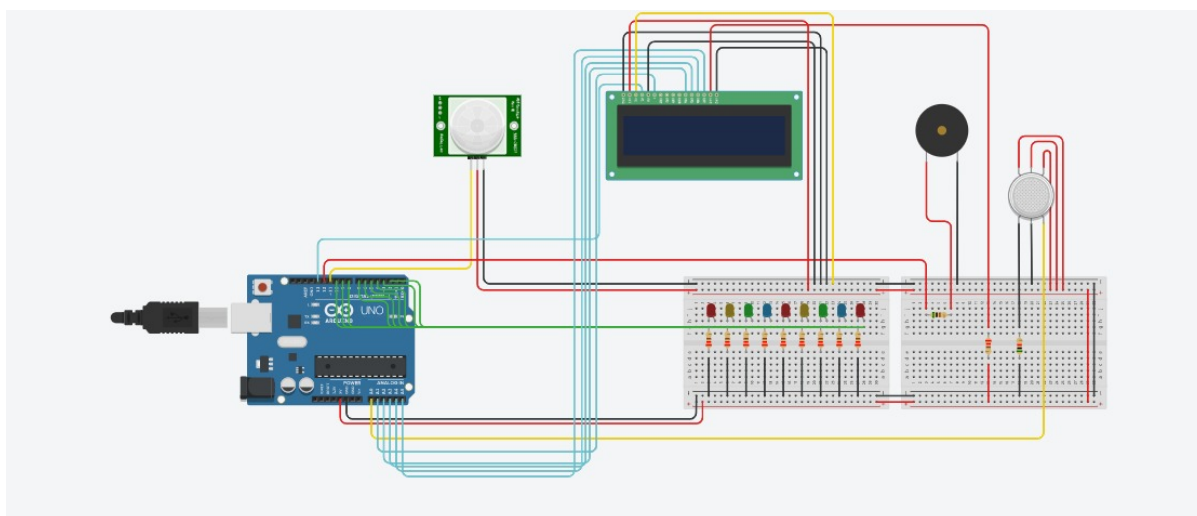
Pentru a realiza acest proiect vom avea nevoie de:

Componente	Numar	Descriere generală	Poza
Placă Uno R3 compatibil Arduino	1	Plăcuță de dezvoltare	
Modul Senzor PIR HC-SR501(Senzor de mișcare)	1	Are o sensibilitate ridicată. Atât raza de acțiune cât și timpul de delay pot fi modificate chiar prin rotirea unor butoane aflate pe componentă. Pentru proiect am ales sensibilitatea cea mai scăzută(3 metri pentru distanță și 0.3 secunde pentru delay)	
Modul Senzor Gaz MQ-2(Senzor de gaz)	1	Folosit pentru a detecta scurgerile de gaze, cu o sensibilitate ridicată. Am folosit citirea de date în mod analog pentru a prelua informațiile în timp real și a semnala cât mai repede prezența gazelor	
LED roșu de 3mm cu Lentile Difuze	9	Sunt folosite pentru a semnala pericolul printr-un semnal luminos. Becurile se aprind și se sting pe rând. Tiparul este mai rapid în cazul scurgerilor de gaze și mai lent în cazul sesizării mișcării	

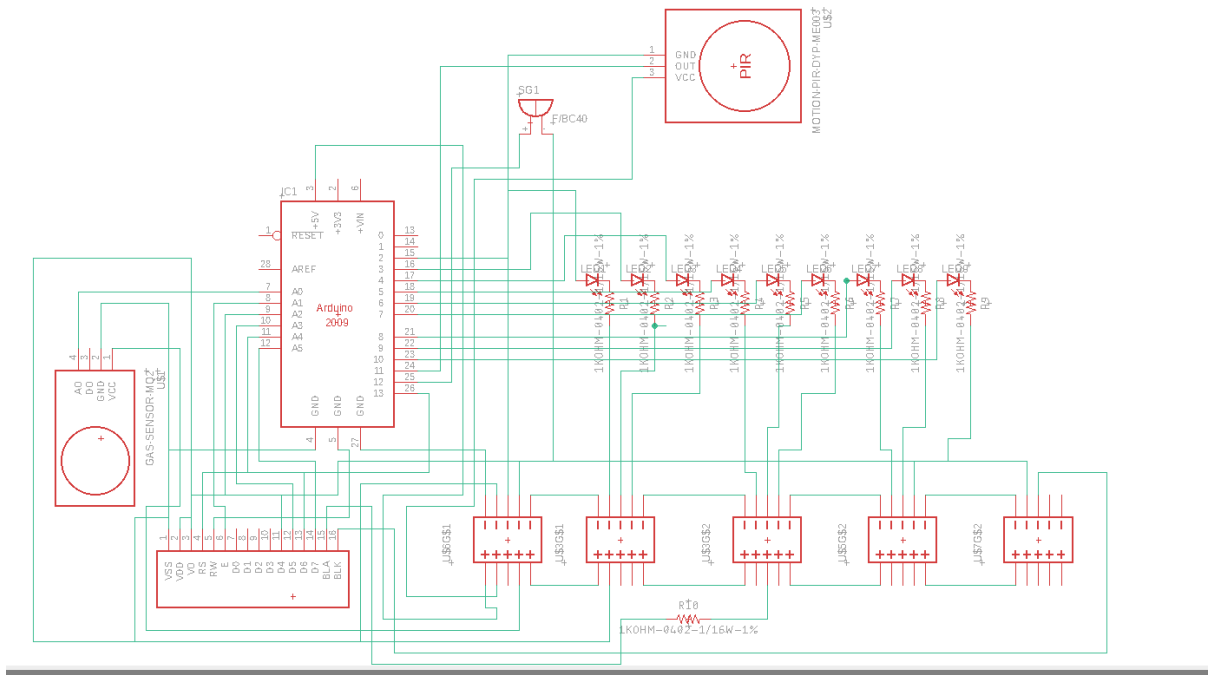
Rezistente 1K	5	Tensiunea oferită de plăcuță este de 5V, dar led-urile funcționează cu o tensiune de alimentare între 2 și 3V. Rezistențele asigură pierderea de tensiune necesară, astfel încât led-urile să nu se ardă	
Breadboard HQ(830 Puncte)	1	Ajută la implementare proiectului fără a fi necesară lipirea componentelor	
LCD 1602 cu Interfață I2C și Backlight Albastru	1	Folosit pentru a afișa mesaje. "ALL SAFE" totul se află sub control. "INTRUDER ALERT" s-a semnalat mișcare și "FIRE FIRE. GAS ALERT" pentru scurgerile de gaz. Mesajul pentru scurgerea de gaz are un grad mai mare de importanță(el va fi afișat în cazul în care ambele sunt activate)	
Buzzer Activ SFM-20B (3-24V)	1	Oferă un semnal sonor(diferit pentru ambele cazuri de alertă)	

Schema electrică:

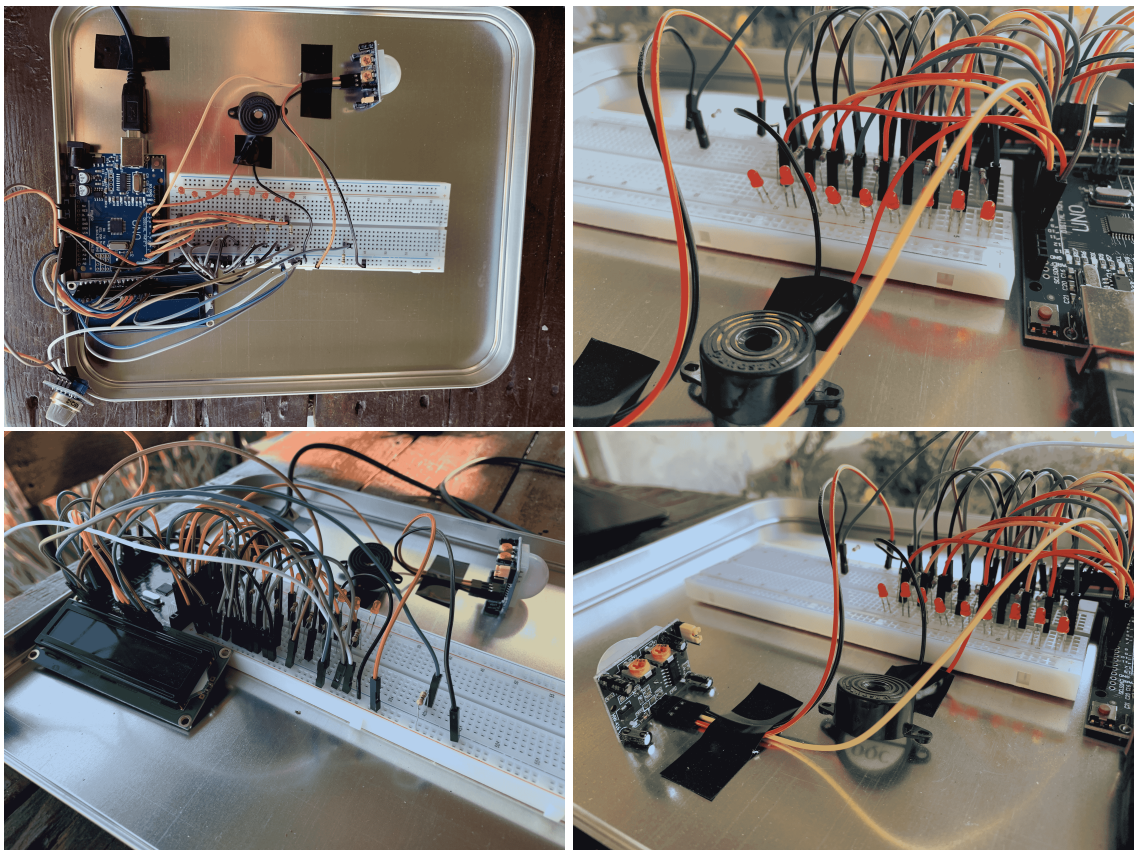
Prima încercare de implementare a proiectului a fost în tinkercad, rezultatul final fiind schema următoare:



După ce am comandat piesele am realizat că senzorul de gaz cumpărat nu se potrivea cu cel folosit și nu exista în tinkercad o altfel de variantă pentru acesta. Astfel, am recreat schema de cablaj în Eagle, rearanjând și poziționarea componentelor, pentru a fi mai ușor de implementat fizic.



În final proiectul arată așa:



Software Design

Codul de implementare este:

```
#include <LiquidCrystal.h>

int pirsensor = 0;
int gassensor = 0;
LiquidCrystal lcd(13, A1, A2, A3, A4, A5);

void setup()
{
  //-----Leds-----
  pinMode(1, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);

  //-----PIR sensor-----
  pinMode(11, INPUT);

  //-----Gas sensor-----
  Serial.begin(9600);

  //-----Piezo-----
  pinMode(12, OUTPUT);

  //-----LCD-----
  lcd.begin(16, 2); // 16 coloane, 2 randuri
}

void loop() {

  pirsensor = digitalRead(11);
  gassensor = analogRead(A0);

  //-----PIR sensor for motion detection-----
  if (pirsensor == HIGH) {
    lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("*INTRUDER*");
    lcd.setCursor(5, 1);
  }
}
```

```
lcd.print("*ALERT*");

// dance led, dance
tone(12, 220, 900);
digitalWrite(2, HIGH);
delay(100);
digitalWrite(2, LOW);
digitalWrite(3, HIGH);
delay(100);
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
delay(100);
digitalWrite(4, LOW);
digitalWrite(5, HIGH);
delay(100);
digitalWrite(5, LOW);
digitalWrite(6, HIGH);
delay(100);
digitalWrite(6, LOW);
digitalWrite(7, HIGH);
delay(100);
digitalWrite(7, LOW);
digitalWrite(8, HIGH);
delay(100);
digitalWrite(8, LOW);
digitalWrite(9, HIGH);
delay(100);
digitalWrite(9, LOW);
digitalWrite(10, HIGH);
delay(100);
digitalWrite(10, LOW);
}
//-----Gas sensor-----
if (gassensor > 650) {
  lcd.clear();
  lcd.setCursor(2, 0);
  lcd.print("*FIRE FIRE*");
  lcd.setCursor(2, 1);
  lcd.print("*GAS ALERT*");

  tone(12, 500, 900);
  digitalWrite(2, HIGH);
  delay(50);
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  delay(50);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  delay(50);
```

```
digitalWrite(4, LOW);
digitalWrite(5, HIGH);
delay(50);
digitalWrite(5, LOW);
digitalWrite(6, HIGH);
delay(50);
digitalWrite(6, LOW);
digitalWrite(7, HIGH);
delay(50);
digitalWrite(7, LOW);
digitalWrite(8, HIGH);
delay(50);
digitalWrite(8, LOW);
digitalWrite(9, HIGH);
delay(50);
digitalWrite(9, LOW);
digitalWrite(10, HIGH);
delay(50);
digitalWrite(10, LOW);
} else {
  // all safe
  lcd.clear();
  lcd.setCursor(5, 0);
  lcd.print("*ALL*");
  lcd.setCursor(5, 1);
  lcd.print("*SAFE*");

  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  digitalWrite(10, LOW);
}

delay(10);
}
```

Am folosit librăria LiquidCrystal pentru a putea controla LCD-ul.

Am ales sa folosesc drept mediu de dezvoltare **Arduino IDE**.

Rezultate Obținute

Funcționalitatea proiectului poate fi văzută aici: [Proiect PM Emilia Cocoru](#).

În final toate obiectivele de la început au fost îndeplinite și implementate cu succes.

Concluzii

Proiectul mi s-a părut extrem de interesant. Au fost momente când nimic nu părea că funcționează, dar toate problemele au fost rezolvate pe rând, câte una la un moment, folosindu-mă de Internet, cât și de ajutor din partea colegilor.

Proiectul a fost implementat pe etape, astfel încât să mă asigur că fiecare componentă funcționează corect. Făcând asta, procesul de debugging a fost destul de simplu de realizat.

Download

Documentatia in format PDF: [Home Security System PDF](#)

Link arhivă date esențiale proiect: [Home Security System Arhivă](#)

Bibliografie/Resurse

Datasheet-uri folosite:

- [Arduino UNO](#)
- [PIR Sensor\(HC-SR501\)](#)
- [MQ2 Gas Sensor](#)
- [I2C LCD Display](#)
- [Buzzer](#)

Diagrama bloc: <https://www.lucidchart.com>

Schema electrică: <https://www.tinkercad.com>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/homesecuritysystem>



Last update: **2021/06/03 10:51**