

Multifunctional Beacon

Introducere

Girofarul multifunctional este menit sa poata fi utilizat atat de echipajele de salvare si de politie, cat si de cele de pompieri, astfel acesta devine un girofar universal, care lumineaza si emite sunete in functie de echipajul care il foloseste. Instrumentul poate fi folosit de echipajele de urgenta in cadrul misiunilor speciale.

Descriere generală

Modul de functionare al proiectului:

- Se alege modul de utilizare prin intermediul unei aplicatii Android (mod salvare, mod politie, mod pompieri)
- Se trimite comanda de la tastatura pentru a alege modul de functionare (din cauza problemelor de conectivitate la modulul Bluetooth, programul este functional doar prin introducerea de date de la tastatura, aplicatia Android fiind partial implementata)
- In functie de modul ales, girofarul va incepe sa lumineze si sa emita sunete corespunzatoare echipajului ales daca sunt vehicule prin preajma
- Pe baza unor senzori de distanta prin care detectam prezenta obstacolelor in fata vehiculului de urgenta, sunetul se reproduce pe masura ce echipajul se apropie de obstacole

Schema bloc a proiectului:



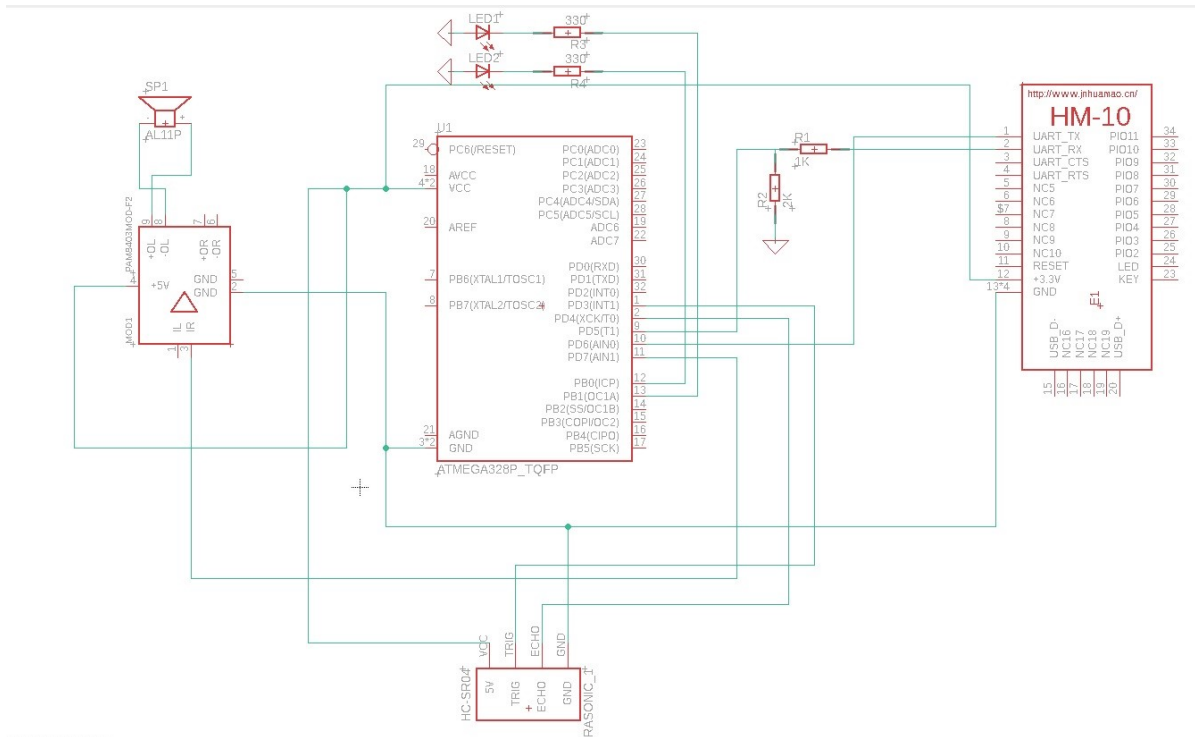
Hardware

Piese utilizate in cadrul proiectului sunt:


- Arduino UNO
- Amplificator
- Difuzor
- LED-uri
- Rezistori
- Breadboard

- Senzor de distanta ultrasonic
- Modul Bluetooth

Schema electrica:



Software

Pentru partea de Arduino, am importat librariile NewPing.h pentru senzorul de distanta si SoftwareSerial pentru modulul bluetooth (didn't work but i tried ).

```
1  #include <NewPing.h>
2  #include <SoftwareSerial.h>
3
4  #define bluetoothRx 6
5  #define bluetoothTx 5
6  #define sensorTrig 9
7  #define sensorEcho 10
8  #define redLed 12
9  #define blueLed 13
10 #define soundSignal 11
11 #define frequency 1000
12 #define maxDistance 20
13
14 SoftwareSerial bluetooth(bluetoothRx, bluetoothTx);
15 NewPing sonar(sensorTrig, sensorEcho);
16
```

Din punct de vedere al complexitatii, codul este unul destul de simplu, avand metodele principale setup() si loop(), insotite de alte 5 metode, playAmbulance(), playPolice(), playFirefighters(), ledOnOff(int ledType), ledsOnOff(int ledType1, int ledType2). Primele 3 metode sunt destinate pentru sunetele ce vor fi emise de sirena, iar ultimele doua sunt functii prin care actionam led-urile sa lumineze intermitent. Functia ledOnOff este implementata pentru ambulanta si pompieri, unde o singura culoare este necesara, iar ledsOnOff este functia pentru politie, unde actionam atat led-ul rosu, cat si pe cel albastru.

```
17 void playAmbulance() {
18     int siren[] = {800, 1200, 1400, 1800, 2000, 2200, 2400, 2800, 3000, 3200, 3400, 3600, 3800, 4000, 4200, 4400};
19
20     for(int i = 0; i < sizeof(siren) / sizeof(int); i++) {
21         tone(soundSignal, siren[i]);
22         delay(150);
23     }
24
25     noTone(soundSignal);
26 }
27
28 void playPolice() {
29     int siren[] = {1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000};
30
31     for(int i = 0; i < sizeof(siren) / sizeof(int); i++) {
32         tone(soundSignal, siren[i]);
33         delay(150);
34     }
35
36     noTone(soundSignal);
37 }
38
39 void playFirefighters() {
40     int siren[] = {200, 400, 600, 800, 1000, 1200, 1400, 1600};
41
42     for(int i = 0; i < sizeof(siren) / sizeof(int); i++) {
43         tone(soundSignal, siren[i]);
44         delay(150);
45     }
46
47     noTone(soundSignal);
48 }
```

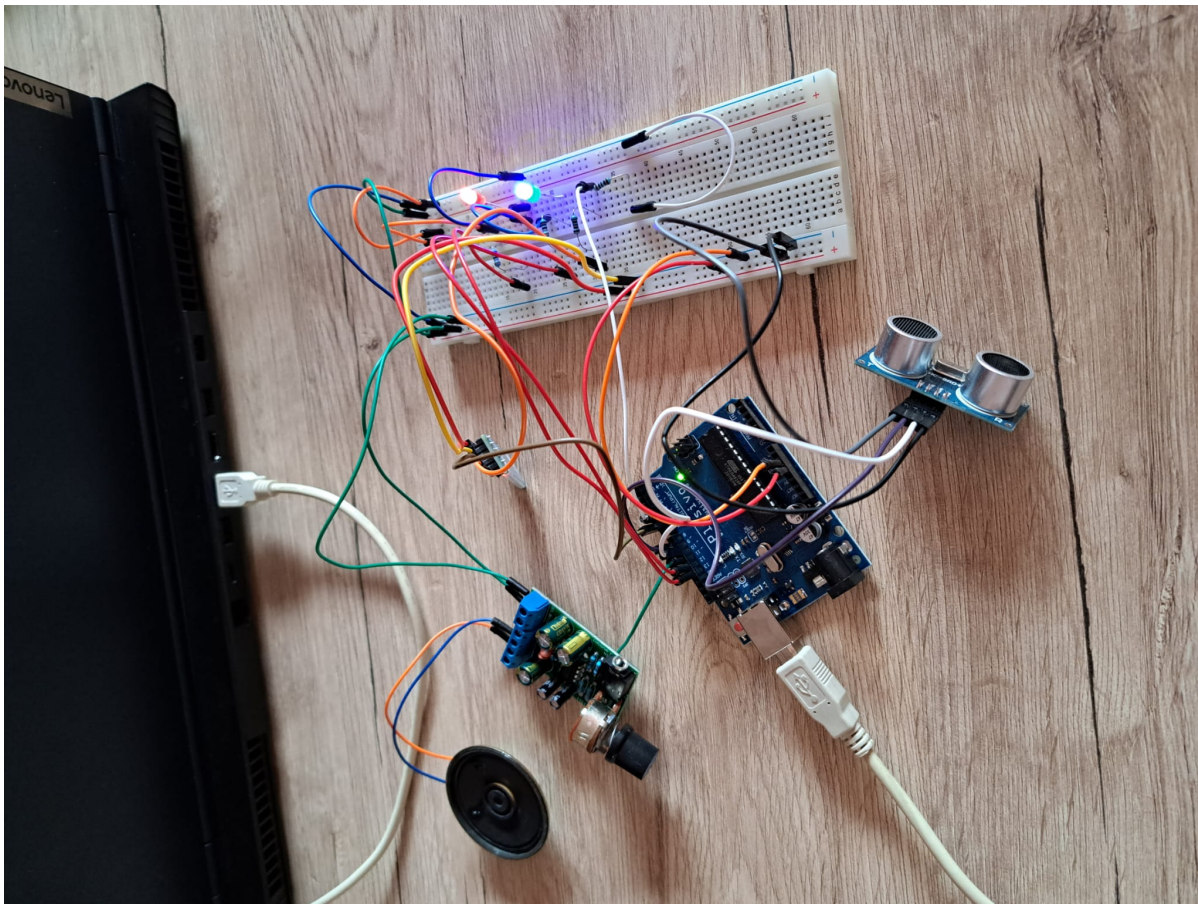
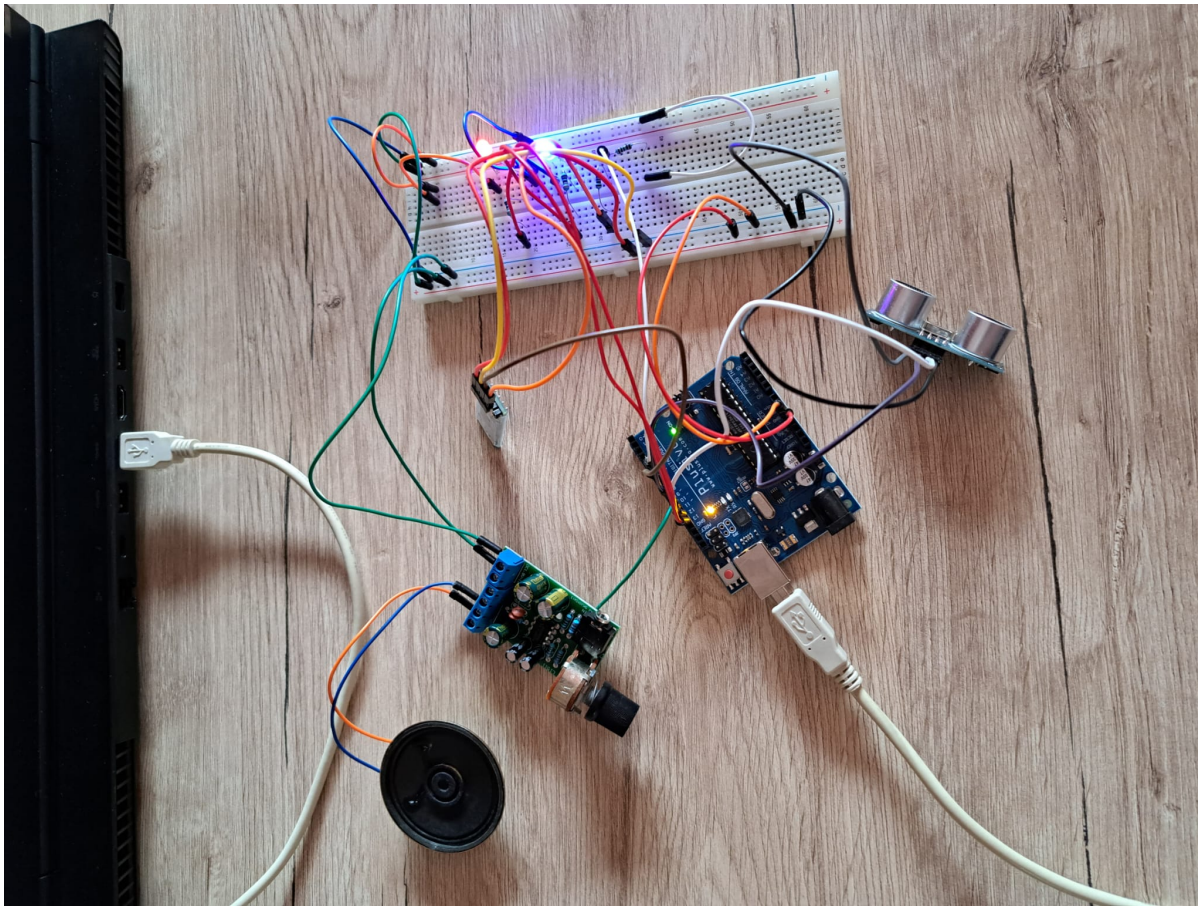
```
50 void ledOnOff(int ledType){
51     for(int i = 0; i < 50; i++){
52         if(i % 5 == 0) {
53             digitalWrite(ledType, HIGH);
54             if(ledType == redLed) {
55                 playFirefighters();
56             } else {
57                 playAmbulance();
58             }
59             delay(100);
60         } else {
61             digitalWrite(ledType, LOW);
62             delay(100);
63         }
64     }
65 }
66
67 void ledsOnOff(int ledType1, int ledType2){
68     for(int i = 0; i < 50; i++){
69         if(i % 5 == 0) {
70             digitalWrite(ledType1, HIGH);
71             digitalWrite(ledType2, HIGH);
72             playPolice();
73             delay(100);
74         } else {
75             digitalWrite(ledType1, LOW);
76             digitalWrite(ledType2, LOW);
77             delay(100);
78         }
79     }
80 }
```

In functia loop retin distanta fata de obstacol si citesc inputul de la tastatura, iar daca am introdus input valid si distanta fata de urmatorul obstacol este mai mica decat distanta maxima stabilita din program, atunci se permite jocul de lumini si emisia sunetelor corespunzatoare modului ales.

```
82 void setup() {
83     // put your setup code here, to run once:
84     pinMode(redLed, OUTPUT);
85     pinMode(blueLed, OUTPUT);
86     pinMode(soundSignal, OUTPUT);
87
88     Serial.begin(9600);
89     bluetooth.begin(9600);
90
91 }
92
93 void loop() {
94     // put your main code here, to run repeatedly:
95     unsigned int distance = sonar.ping_cm();
96
97     if(Serial.available()) {
98         char input = Serial.read();
99
100        if(input == '0' && distance <= maxDistance) {
101            ledOnOff(blueLed);
102            digitalWrite(redLed, LOW);
103        } else if (input == '1' && distance <= maxDistance) {
104            ledsOnOff(redLed, blueLed);
105        } else if (input == '2' && distance <= maxDistance) {
106            ledOnOff(redLed);
107            digitalWrite(blueLed, LOW);
108        }
109
110        while(Serial.available()) {
111            Serial.read();
112        }
113    }
114
115 }
```

In cadrul aplicatiei Android, am realizat design-ul aplicatiei, destul de simplist, cu 3 butoane care vor trimite date catre modulul bluetooth, dar din cauza conectivitatii esuate la modulul bluetooth, nu am finalizat aplicatia.

Rezultate Obtinute



Surse

[332cc_nechifor_tudor_multifunctional_beacon.zip](#)

Bibliografie/Resurse

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/alucaci/multifunctional-beacon>



Last update: **2023/05/28 15:20**