

# Parking Sensor

## Introducere

Prezentarea pe scurt a proiectului vostru:

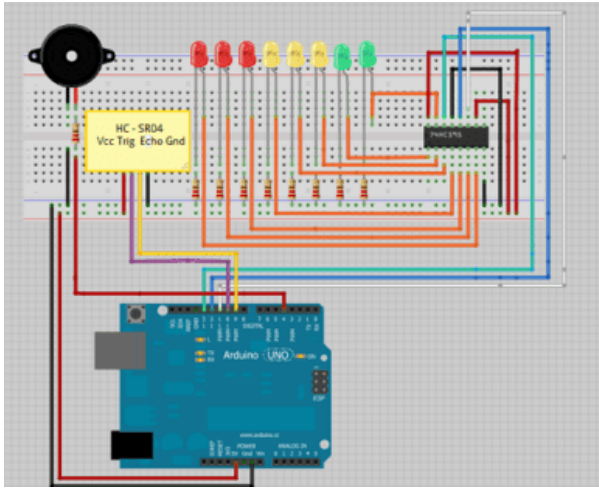
- un senzor ultrasonic care măsoară distanța și graficul bară cu LED-uri se va aprinde în funcție de distanța noastră de la senzor
- poate funcționa ca un senzor de parcare
- nevoia de ajutor la parcare cu spatele
- asistenta la parcare pentru a evita orice fel de contact

## Descriere generală

## Hardware Design

Aici puneți tot ce ține de hardware design:

- Arduino UNO
- 1x 74HC595 - registru de deplasare pe 8 biți
- 1x Breadboard
- LED-uri 8x (de exemplu: 3x roșu, 3x galben, 2x verde)
- Rezistențe de 9 Ohm 220 Ohm
- 1x Buzzer
- 1x senzor ultrasonic (de exemplu: HC-SR04)
- Jumper Wires



## Software Design

```
int tonePin = 4;    //Tone - Red Jumper
int trigPin = 9;   //Trig - violet Jumper
int echoPin = 10;  //Echo - yellow Jumper
int clockPin = 11; //IC Pin 11 - white Jumper
int latchPin = 12; //IC Pin 12 - Blue Jumper
int dataPin = 13;  //IC Pin 14 - Green Jumper

byte possible_patterns[9] = {
  B00000000,
  B00000001,
  B00000011,
  B00000111,
  B00001111,
  B00011111,
  B00111111,
  B01111111,
  B11111111,
};
int proximity=0;
int duration;
int distance;

void setup() {
  //Serial Port
  Serial.begin (9600);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(latchPin, OUTPUT);
```

```
pinMode(dataPin, OUTPUT);
pinMode(tonePin, OUTPUT);
}

void loop() {
  digitalWrite(latchPin, LOW);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(1000);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  /*if (distance >= 45 || distance <= 0){
    Serial.println("Out of range");
  }
  else {
    Serial.print(distance);
    Serial.println(" cm");
  }*/

  proximity=map(distance, 0, 45, 8, 0);
  //Serial.println(proximity);

  if (proximity <= 0){
    proximity=0;
  }
  else if (proximity >= 3 && proximity <= 4){
    tone(tonePin, 200000, 200);
  }
  else if (proximity >= 5 && proximity <= 6){
    tone(tonePin,5000, 200);
  }
  else if (proximity >= 7 && proximity <= 8){
    tone(tonePin, 1000, 200);
  }
  shiftOut(dataPin, clockPin, MSBFIRST, possible_patterns[proximity]);
  digitalWrite(latchPin, HIGH);


  delay(600);
  noTone(tonePin);
}
```

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## Concluzii

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).  
**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

## Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

## Bibliografie/Resurse

[Documentatie](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/apredescu/alarmclock> 

Last update: **2022/06/02 09:32**