**Realizarea Senzorilor de Proximitate cu Arduino Mega 2560**

**Proiect Sisteme Încorporate**

**INTRODUCERE:**

Realizarea Senzorilor de Proximitate cu Arduino Mega 2560 este un proiect realizat în cadrul cursului universitar de Sisteme Încorporate care își dorește să ajute la aporximarea distanței dintre obstacolele întâlnite de către un vehicul sau alt corp în mișcare.

Deoarece proiectul este inspirat din sistemele de parcare convenționale ale autovehiculelor din zilele noastre, acesta își propune să rezolve problema mașinilor care nu dispun de un astfel de sistem încorporat care să ajute conducătorul auto în realizarea diverselor manevre, în special cele de parcare(parcare cu fața , parcare cu spatele și parcare laterală ).

Mulți șoferi începători și nu numai, întâmpină probleme atunci când vine vorba despre parcarea în diferite moduri. Sistemele de proximitate pentru parcarea pe mașini au început să apară pe scară largă în anii 2000, cu unele prime implementări mai limitate și experimentale datând încă din anii '90. Aceste sisteme au devenit treptat standard pe multe mașini mai noi și sunt acum parte integrantă a multor tehnologii avansate de asistență pentru șoferi și siguranță a vehiculelor.

Aceste sisteme au apărut întocmai pentru ajutorarea șoferilor auto pentru aproximarea distanței dintre autovehicul și obstacole sau alte autovehicule. Apariția acestor sisteme încorporate a început astfel să limiteze numărul de daune ale unei mașini atunci când vine vorba de aptitudinile șoferului de a parca.

**Prezentarea soluției propuse:**

În epoca tehnologiei în care trăim, parcarea devine o provocare tot mai presantă pentru șoferi. Lipsa de spațiu și obstacolele neașteptate pun în pericol mașinile și îngreunează procesul de parcare, punându-ne la încercare abilitățile și răbdarea.

În fața acestor dificultăți, propun implementarea unui sistem încorporat care preia date din mediul înconjurător cu ajutorul unor senzor ultrasonic, detector de distanta de tipul HC-SR04 și care cu ajutorul unei plăcuțe de dezvoltare Arduino Mega 2560, afișează pe un modul OLED, pe arhitectură SSD1306, distanța dintre senzorii de proximitate și autovehicul, într-un design de tip 8‑Bit intuitiv. Pe lângă afișajul OLED, conducătorul auto este ajutat și de către semnalele atât luminoase cât și audio transmise de 3 LED-uri RGB și un Buzzer Activ, în funcție de distanța la care se află față de obstacol.

De asemenea, este important de menționat și aspectul practic al acestui sistem. Folosind componente accesibile și ușor de integrat, sistemul poate fi implementat fără dificultate pe o gamă largă de vehicule, aducând beneficii tangibile tuturor șoferilor, indiferent de experiența lor de condus.

**Descriere sistem:**

Sistemul Încorporat prezentat mai sus are o arhitectură alcătuită atât din module și componente hardware diverse, cât și o parte software. Ambele elemente au o contribuție egală, sistemul neputând merge și oferind rezultatele dorite dacă una dintre acestea nu ar funcționa corect.

Componenta hardware este reprezentată de o listă de componente care are fiecare rolul ei și care comunică între ele pentru rularea în modul dorit a programului.

Lista completă a componentelor hardware utilizate este următoarea:

* 1x Plăcuța de dezvoltare Arduino Mega 2560
* 3x Senzori ultrasonici HC-SR04
* 1x OLED Display 0.96 inch SSD1306
* 1x Buzzer Activ
* 3x LED RGB
* 9x Rezistente de 220 ohm
* 20x Fire de tip jumper

Pentru o ușoară asamblare și încorporare a sistemului s-au folosit 2 plăcuțe de tip Breadboard.

Cei trei senzori ultrasonici măsoară cu ajutorul ultrasunetelor distanța până la un anumit obstacol. Acestea funcționează printr-un sistem de tip "transceiver-receiver". Transceiver-ul care este reprezentat de un pin numita "Trigger", trimite un semnal ultrasonic care atunci când se lovește de un obstacol, sunetul ultrasonic se întoarce și este preluat de un al doilea pin numită "Echo" care are rol de receiver. Informațiile preluate de Echo sunt apoi transformate de către componenta software în centimetri și afișate pe portul serial.

Datele sunt apoi transmise către plăcuța de dezvoltare Arduino Mega cu ajutorul firelor jumper care sunt legate pe diverși pini ai plăcuței. Arduino Mega este practic creierul întregului sistem. Preia și transmite date, alimentează și programează componentele diverse ale sistemului încorporat.

Cu ajutorul componentei software datele preluate de la cei 3 senzori ultrasonici sunt trimise către un afișaj de tip OLED prin 2 pini de tip SCL și SDA pe care este afișată o interfața de măsurare aproximativă a distanței către obstacol prin 4 linii așezate pe câte 3 coloane. În funcție de distanța la care sistemul încorporat se află față de obstacol, cele 4 linii sunt pline, dacă obiectul este departe, și hașurate, dacă obstacolul este aproape.

Afișajul OLED este dublat de 3 LED-uri RGB și un Buzzer Activ. LED-urile își schimbă culorile în funcție de distanța la care se află obstacolul față de senzorul ultrasonic cu care este asociat. Verde, reprezintă distanța la care autovehiculul se afla la cea mai sigură distanță față de obstacol, fiind urmat de galben, roșu, iar dacă obstacolul se află prea aproape, LED-ul RGB va deveni alb.

Buzzer-ul Activ funcționează pe același prinicpiu ca și cele 3 LED-uri RGB, doar că acesta folosește semnale de tip sonor care sunt din ce în ce mai dese, o dată cu scăderea de distanța dintre obiect și sistemul încorporat. Pe lângă asta, buzzer-ul activ este atribuit tuturor celor 3 senzori ultrasonici, declanșându-se astfel indiferent de senzorul care este activat.

Cele 9 rezistențe cu 5 benzi a câte 220 ohm au rolul de a proteja cele 3 LED-uri RGB

**Componente Hardware:**

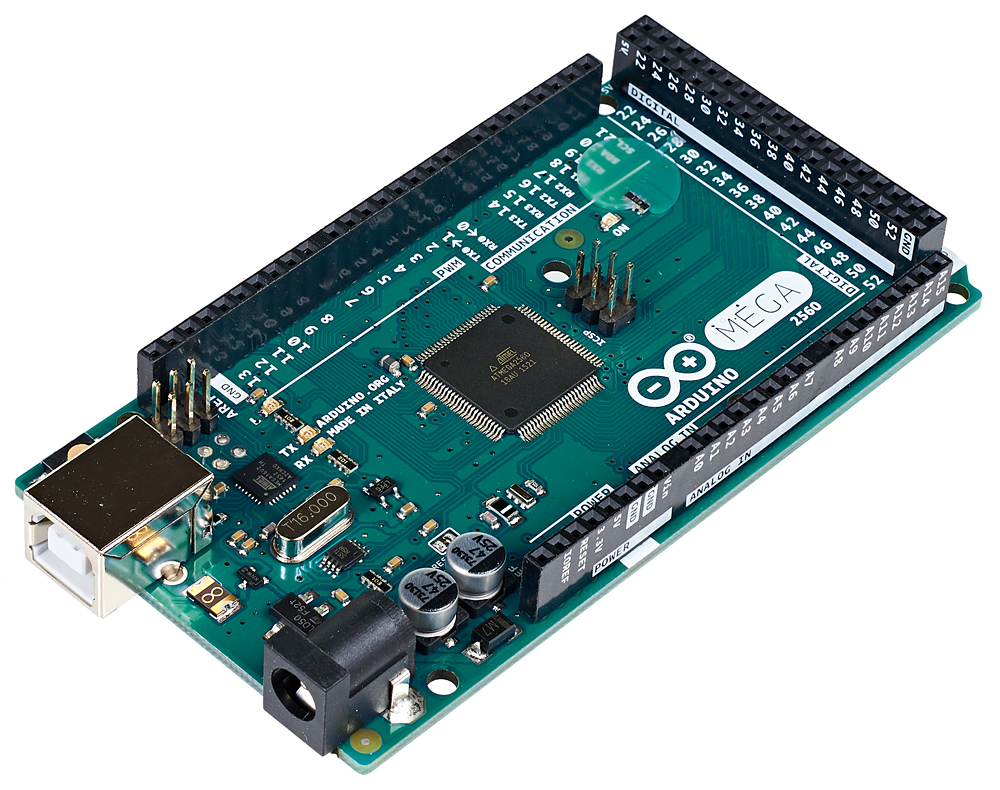
**Arduino Mega 2560:**

Arduino Mega 2560 este o placă de dezvoltare microcontroler bazată pe ATmega2560. Este o placă populară aleasă pentru proiecte complexe care necesită un număr mare de pini I/O și memorie flash.

**Caracteristici principale:**

* **Microcontroler:** ATmega2560 cu o frecvență de 16 MHz
* **Memorie:** 256 KB flash, 8 KB SRAM, 4 KB EEPROM
* **Pini I/O:** 54 pini digitali (din care 15 cu PWM) și 16 pini analogici
* **Conectivitate:** USB, 4 UART-uri (seriale)
* **Alte caracteristici:** Soclu ICSP pentru programare in-circuit, buton de resetare, LED de alimentare

**Documentația Arduino Mega 2560:** <https://docs.arduino.cc/hardware/mega-2560>



**Afișaj OLED SSD1306 0.96 inch:**

Afisajul OLED SSD1306 0.96 inch este un modul LCD cu rezolutie 128x64 pixeli, bazat pe tehnologia OLED. Este o solutie compacta si eficienta pentru afisarea informatiilor text si grafice simple, fiind apreciata pentru consumul redus de energie si contrastul ridicat.

**Caracteristici principale:**

* **Dimensiune:** 0.96 inch (diagonală)
* **Rezolutie:** 128 x 64 pixeli
* **Tehnologie:** OLED (Organic Light Emitting Diode)
* **Interfata:** I2C (Comunicare in serie cu 2 fire)
* **Tensiune de alimentare:** 3.3V - 5V
* **Consum de energie:** Foarte redus (ideal pentru proiecte portabile)
* **Contrast:** Ridicat, cu o vizibilitate excelenta chiar si in lumina directa a soarelui
* **Unghi de vizualizare:** Larg, permitand vizualizarea afisajului din diverse unghiuri
* **Durabilitate:** Foarte buna, cu o durata de viata lunga

**Documentația SSD1306:** <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>



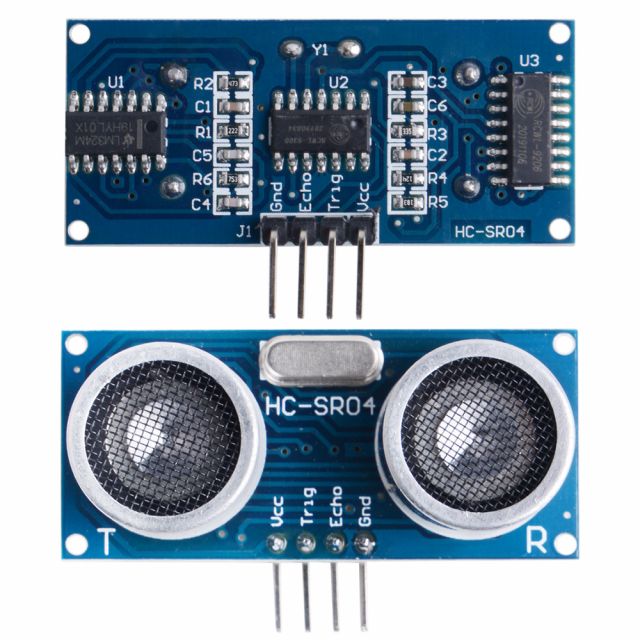
**Senzorul ultrasonic HC-SR04:**

Senzorul ultrasonic HC-SR04 este un dispozitiv electronic mic, dar puternic, utilizat pentru a măsura distanța până la obiecte. Functioneaza similar unui sonar folosit de lilieci sau submarine, emitand unde sonore cu frecvență înaltă și captând ecoul reflectat de obiecte.

**Caracteristici principale:**

* **Tehnologie:** Ultrasunete
* **Domeniu de măsurare:** 2 cm - 400 cm (poate varia in functie de specificatii)
* **Precizie:** Până la 3 mm
* **Interfata:** 4 pini (VCC, GND, Trig, Echo)
* **Tensiune de alimentare:** 5V DC
* **Curent de operare:** 15 mA (aproximativ)
* **Unghi de detecție:** ~30 grade

**Documentația HC-SR04:** <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132204/ETC2/HCSR04.html>



**LED RGB:**

LED-ul RGB (Red, Green, Blue) este un tip special de LED capabil să emită o gamă largă de culori prin combinarea intensității luminii diodelor individuale roșu, verde și albastru. Spre deosebire de LED-urile simple care emit o singură culoare, LED-ul RGB oferă o paleta vastă de posibilități prin controlul independent al intensității fiecărui LED intern.

**Caracteristici principale:**

* **Multicolor:** Poate emite o gamă largă de culori prin combinarea luminii roșu, verde și albastru.
* **Control individual:** Intensitatea fiecărui LED intern (roșu, verde, albastru) poate fi controlată independent.
* **Dimensiune compactă:** Disponibil în diferite dimensiuni, putând fi montat pe o placă de proiectare sau integrat în dispozitive.
* **Eficient energetic:** Consumă mai puțină energie decât becurile incandescente tradiționale.
* **Durata mare de viață:** Poate funcționa mii de ore, fiind o soluție durabilă.



**Buzzer Activ:**

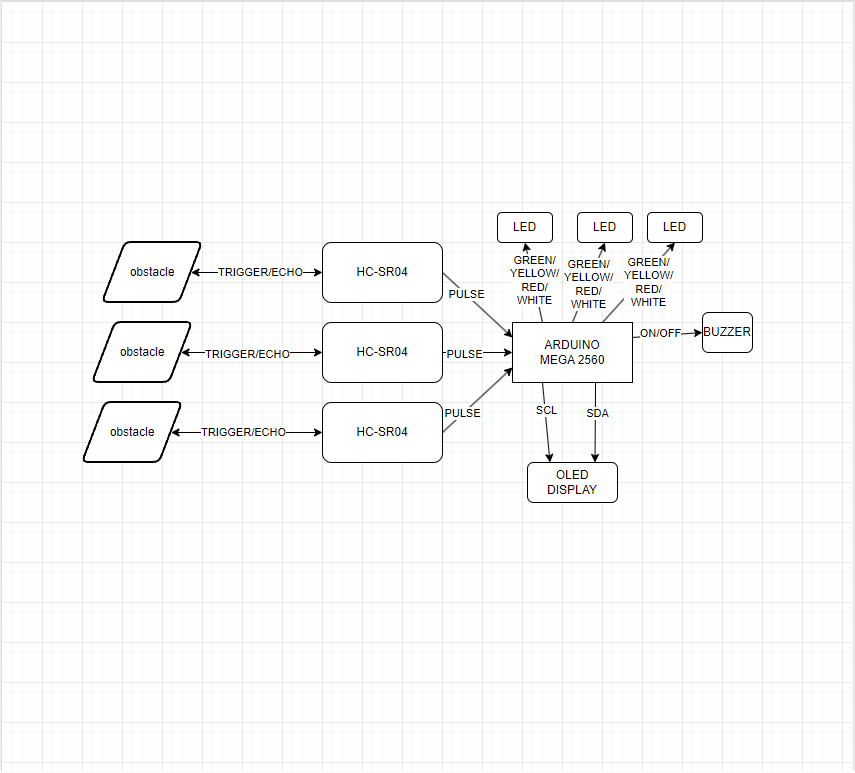
Un Buzzer Activ este un dispozitiv electronic compact care emite un sunet continuu sau pulsat atunci când este alimentat cu energie electrică. Spre deosebire de un difuzor care reproduce sunete complexe, un Buzzer Activ generează un sunet electronic simplu, similar unui bip sau unui zumzet.

**Caracteristici principale:**

* **Sunet electronic:** Produce un sunet continuu sau pulsat cu o tonalitate simplă.
* **Compact:** Dimensiuni reduse, putând fi montat cu ușurință pe o placă de proiectare.
* **Tensiune de alimentare:** Functioneaza la o tensiune specificata, de obicei între 3V si 12V DC (curent continuu).
* **Volum reglabil:** Unele buzzere permit reglarea volumului prin modificarea tensiunii de alimentare sau prin metode electronice.
* **Ieftin:** O componentă electronică accesibilă pentru majoritatea proiectelor.



**DIAGRAMA BLOC:**



**Componente Software:**

* Transformarea imaginilor în matice de octeți pentru utilizarea lor pe display-ul monocrom OLED
* Definirea pinilor pentru cei 3 senzori ultrasonici
* Definirea numărului de senzori (3)
* Definirea pinilor pentru buzzer și led-urile rgb
* Gruparea variabilelor atribuite senzorilor într-un struct
* Adăugarea unei matrice de dimensiunea numărului senzorilor în struct
* Inițializarea a două variabile de distanță minimă și maximă (2 cm și 400 cm)
* Inițializarea a 4 variabile care reprezintă praguri în cm a modificărilor pe display-ul OLED, LED-urile RGB și Buzzer în funcție de distanța la care se află senzorii față de obiect
* Inițializarea stării buzzerului pe starea LOW
* Inițializarea unor variabile pentru contorizarea timpului și intervale de tip pentru zumzetul buzzer-ului
* Crearea a 4 funcții Buzz care funcționează pe același principiu, doar că la intervale de timp diferite:

currentTime-previousBuzz > buzzPeriod -> a trecut o anumită perioadă de timp de la ultima activare a buzzer-ului

currentTime -> timpul curent contorizat de la pornirea plăcuței

PreviousBuzz -> stochează timpul când a fost buzzer-ul ultima dată activat

buzzPeriod -> intervalul de timp la care buzzer-ul se activează

If (currentTime-previousTime > buzzPeriod) -> verifică dacă, condiția este adevărata pentru executarea condiției

previousBuzz currentTime -> actualizează variabila previousBuzz cu timpul curent pentru a urmări cel mai recent punct de activitate

If (buzzerState LOW) {buzzerState HIGH}else{buzzerStateLOW}->comutarea buzzer-ului din starea LOW în starea HIGH

* Atribuirea fiecărui cu pinii de trigger si echo
* Setarea pinilor ca ieșiri și intrări
* Setarea fontului si a culorii fontului pe OLED
* Pornirea portului serial
* Atribuirea variabilei currentTime cu funcția millis care pornește ceasul si numără cât timp a trecut de la pornirea plăcuței
* Crearea unei bucle for pentru măsurarea continuă a celor 3 senzori
* Afișarea valorilor transformate în cm pe portul serial(pentru debug)
* Crearea unei condiții if pentru a apela cele 4 funcții care fac buzzer-ul să bipăie la intervale de timp diferite
* If(sensor[i].distanta\_cm >= dist01)-> dacă această condiție este îndeplinită, se apelează funcția de buzz
* Crearea unor condiții imbricate pentru setarea culorilor led-urilor în funcție de distanța la care se află obstacolul
* if(sensor[1].distanta\_cm >= dist01) -> dacă această condiție este îndeplinită, primul led își setează culoarea verde pe HIGH, iar restul culorilor pe LOW
* u8g.firstPage()-> Inițializează afișarea pe prima pagină.
* do{ ... }while(u8g.nextPage() )-> această buclă se repetă atâta timp cât există pagini disponibile pentru afișare.
* În interiorul buclei se execută forme scurte de if, folosind operatroul ternar care afișează imaginile cu liniile sugestive pentru interpretarea distanței față de obstacol.
* Acestea sunt afișate la distanțe date față de partea din stânga și partea de sus a ecranului OLED.
* Dacă senzorul are o valoarea mai mare decât cea nominală atribuită mai devreme în program, vă afișa o linie plină (true), altfel va afișa o linie punctată (false).
* De asemenea mai sunt afișate și unitatea de măsură (cm) în partea de sus spre stânga și un indicator care să denote faptul că există și semnal audio în partea de sus spre dreapta

**IMAGINI DE PREZENTARE:**