



**TECHNICAL  
UNIVERSITY**  
OF CLUJ-NAPOCA  
ROMANIA

**FACULTATEA: Automatică și Calculatoare**  
**SPECIALIZAREA: Calculatoare și Tehnologia Informației**  
**DISCIPLINA: Proiectare cu microprocesoare**  
**PROIECT: Radar Arduino**

**Îndrumător laborator:**

**Mircea Paul Muresan**

**Student:**

**Galiș George-Laurențiu**



## Cuprins

<b>1. Specificatii si analiza proiectului.....</b>	<b>3</b>
<b>2. Precizarea limbajului de programare alese.....</b>	<b>3</b>
<b>3. Componentele folosite si proiectarea acestora.....</b>	<b>4</b>
<b>4. Testarea propriu-zisă a proiectului.....</b>	<b>8</b>
<b>5. Concluzii și dezvoltări ulterioare .....</b>	<b>9</b>



## 1. Specificatii si analiza proiectului

Proiectul de fata reprezinta crearea si folosirea unui detector de obiecte care poate fi numit radar. Acest automat poate detecta obiectele aflate la o anumita distanta si la un anumit unghi fata de senzorul care determina existenta obiectelor. Toate aceste informatii trimise de senzori sunt primite de microcontrolerul aflat pe o placuta Arduino care acestea sunt trimise la calculator prin portul serial si folosite intr-o aplicatie.

Aplicatia creaza o interfata grafica care poate fi inteleasa mult mai usor de catre utilizator aceasta afisand unghiul la care se afla un obiect si distanta la care se afla un obiect masurata in cm. Aplicatia are la baza limbajul de programare Java.

Componentele care sunt folosite de placuta Arduino sunt: un servomotor, un senzor ultrasonic, fire si un breadboard.

## 2. Precizarea limbajului de programare ales

### 2.1. Java

Java este un limbaj de programare orientat-obiect, puternic tipizat, conceput de către James Gosling la Sun Microsystems (acum filială Oracle) la începutul anilor '90, fiind lansat în 1995. Cele mai multe aplicații distribuite sunt scrise în Java, iar noile evoluții tehnologice permit utilizarea sa și pe dispozitive mobile, spre exemplu telefon, agenda electronică, palmtop etc. În felul acesta se creează o platformă unică, la nivelul programatorului, deasupra unui mediu eterogen extrem de diversificat. Acesta este utilizat în prezent cu succes și pentru programarea aplicațiilor destinate intranet-urilor.

Limbajul împrumută o mare parte din sintaxă de la C și C++, dar are un model al obiectelor mai simplu și prezintă mai puține facilități de nivel jos. Un program Java compilat, corect scris, poate fi rulat fără modificări pe orice platformă care e instalată o mașină virtuală Java (engleză Java Virtual Machine, prescurtat JVM). Acest nivel de portabilitate (inexistent pentru limbaje mai vechi cum ar fi C) este posibil deoarece sursele Java sunt compilate într-un format standard numit cod de octeți (engleză byte-code) care este intermediar între codul mașină (dependent de tipul calculatorului) și codul sursă.

### 2.2. C

C este simplu deoarece numărul de componente al limbajului este relativ mic. Spre exemplu dacă două funcții ale limbajului realizează mai mult sau mai puțin același lucru, C va include doar o funcție. **C are anumite avantaje și dezavantaje:** • este extrem de flexibil, o secventa

putând fi codificată în mai multe moduri; aşadar putem scrie acelaşi program în mai multe feluri, el executând aceiaşi funcţie. • permite o scriere compactă, instrucţiunile putând fi scrise pe acelaşi rând; însă acest lucru nu este recomandat deoarece corectarea programului se face mai greu iar după o perioadă nici creatorul nu îşi mai înţelege propriul cod. • programul poate fi scris într-o manieră apropiată de pseudocod (gr.pseudo=fals), conceperea algoritmilor fiind mult mai uşoară şi eficientă. • permite scrierea a aproape orice, greşelile de sintaxă fiind iminente.

C este folosit în general în proiecte de mici dimensiuni unde se pune accent pe performanţa codului. Ex. Programarea microcontrolerelor. În orice caz limbajul C este un limbaj de programare foarte influent şi în acelaşi timp foarte popular. Pentru a realiza un program în C este nevoie de un program pentru scrierea codului (editor de text) şi de un compilator. Compilatorul este un software care transformă codul scris în C (limbaj înţeles de noi) în cod binar (limbaj înţeles de calculator). Compilarea unui program constă în verificarea codului (greşeli de sintaxă), transformarea lui în cod binar şi crearea unui executabil sau a unui fişier în urma compilării programului.

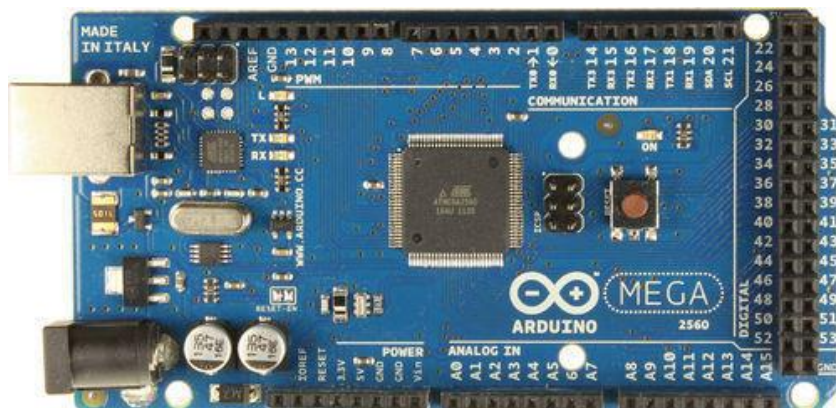
Orice program C este alcătuit din una sau mai multe funcţii. Un program C poate fi asemănat cu o clădire de birouri. Avem biroul fiecărui angajat şi avem biroul şefului. Numărul de birouri ale angajaţilor este nelimitat, fiecare reprezentând o anumită funcţie, iar biroul şefului este numai unul singur el coordonând funcţiile subalternilor. Biroul şefului din exemplul anterior este funcţia **main**, ea fiind nelipsită din orice program C, execuţia programului începând automat cu ea.

### 3. Componentele folosite si proiectarea acestora

#### 3.1. Componente

- **Placuta Arduino Mega**

Familia de unelte de dezvoltare Arduino include plăci cu microcontroller, accesorii şi componente software open source, care permit utilizatorilor să realizeze proiecte folosind o abordare unificată, de nivel înalt, care se doreşte a fi independentă de microcontrollerul folosit. Plăcile Arduino sunt echipate în principal cu microcontrollere Atmel AVR, dar există şi plăci echipate cu microcontrollere de tip ARM, sau din familia x86.



- **Servomotor**

Motoarele servo sunt folosite pentru a obține rotații parțiale, stabile și controlate, pentru efectuarea unor operații cu amplitudine mică dar cu precizie ridicată: acționare mecanism de închidere-deschidere, poziționare senzori, efectuarea unor gesturi, etc. Acesta este folosit la rotirea senzorului pentru o acoperire cât mai mare a suprafeței.



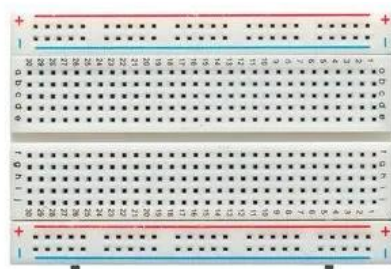
- **Senzor ultrasonic**

Senzorii ultrasonici detectează o multitudine de obiecte, fără contact și fără uzură, cu unde ultrasonice. Unul dintre difuzoare generează o undă ultrasonică (trig), aceasta lovind obiectul și fiind receptată de celălalt senzor (echo).



- **Breadboard**

Breadboard-ul este folosit la extinderea planului de proiectare, mai exact un semnal poate fi folosit în mai multe locuri fiind transmis cu ajutorul firelor.



- **Fire**

Firele sunt folosite la conectarea componentelor intre ele si transmiterea semnalelor.



### 3.2. Proiectarea componentelor

Pentru realizarea acestui proiect am folosit urmatoorii pini de pe placuta Arduino

- Pinii 6 si 7 – pentru semnalele tring respectiv echo de la senzorul ultrasonic
- Pinii 8 – pentru semnalul de comanda de la motorul servo
- Pinul GND – pentru conectarea la masa a componentelor
- Pinul Vin(de 5v) – pentru alimentarea componentelor

Pentru legarea senzorului ultrasonic de motorul servo am folosit niste curele de plastic si motorul servo poate fi mutat, alaturi de placuta, oriunde si poate incepe sa scaneze suprafata respectiva

- **Codul scris in C**

Pentru a face functional motorul servo este nevie de declararea libreriei Servo.h si a unui obiect (se ex **Servo MyServo;**) care foloseste semnalul fizic de control al motrasului. Specificarea pinului folosit pe placuta se face cu ajutorul functiei attach (**myServo.attach(servoPin);**) unde servoPin este numarul pinului folosit. Pozitionarea motorasului la un anumit unghi se face cu ajutorul functiei write(**myServo.write(ungle);**).

Pentru a face functional senzorul ultrasonic mai intai trebui sa legam pinii de pe placuta cu ajutorul metodei pinMode (**pinMode(trigPin, OUTPUT);** **pinMode(echoPin, INPUT);**). trigPin este semnal de output deoarece acesta genereaza ultrasunetul si echoPin este semnal de input deoarece senzorul recepteaza

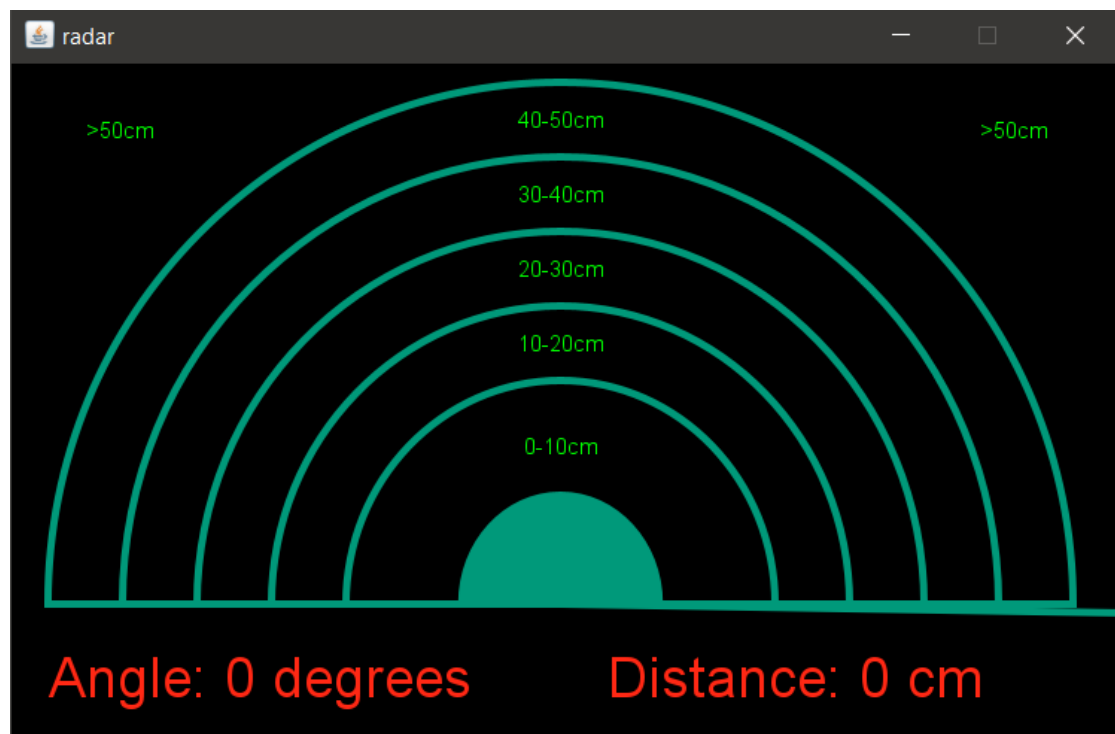
ultrasunetul. In cod, informatia data de semnalul trigPin este calculata cu ajutorul functiei **pulseIn** atunci cand este pe LOW sau pe HIGH (**digitalWrite(trigPin, HIGH);**). Distanța obiectului este calculata cu ajutorul formulei

$$\text{distanța} = \text{timp} * \text{vitezaSunetului} / 2.$$
 (este impartit la 2 deoarece sunetul face un drum dus-intors)

- **Codul scris in Java**

Pentru conexiunea seriala si citirea datelor de pe portul serial se va folosi biblioteca **com.fazecast.jSerialComm.\***; care a fost descarcata prin extensia Maven din mediul de programare IntelliJ. Cu ajutorul metodelor din aceasta biblioteca s-a putut face citirea datelor transmise din aplicatia Arduino in aplicatia Java: distanta si unghiul. Datele au putut fi citite caracter cu caracter cu ajutorul metodei **read()**.

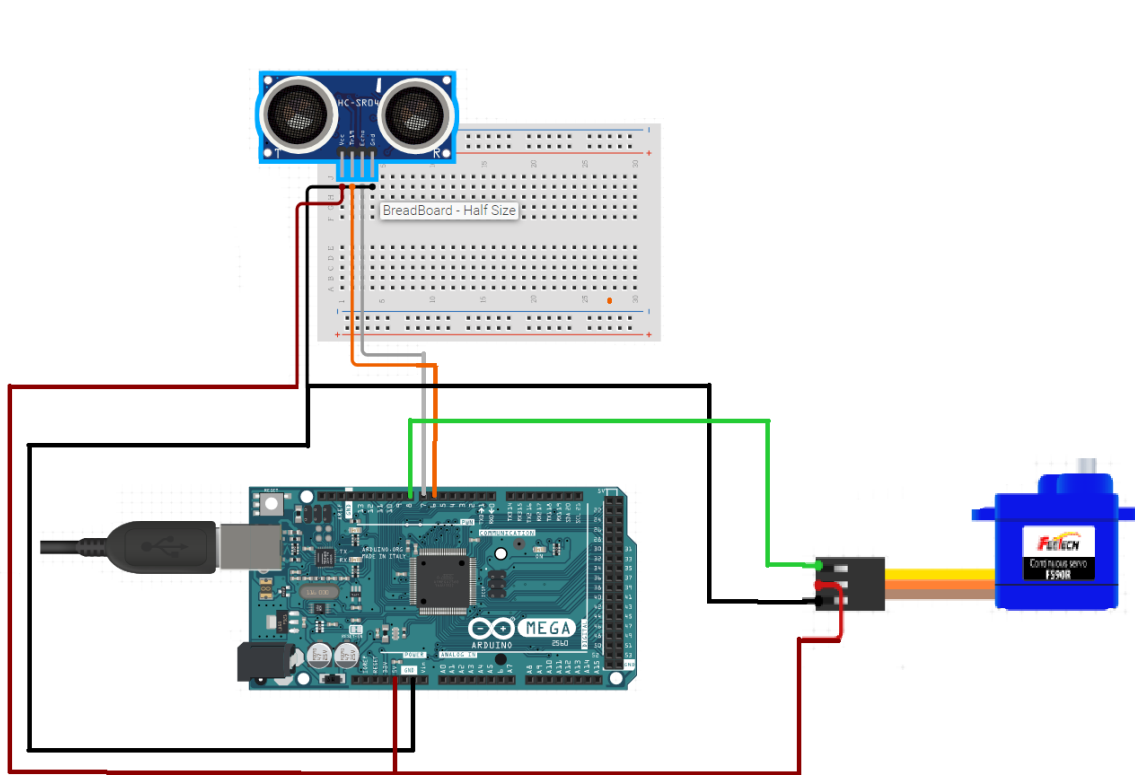
Interfata grafica este realizata cu ajutorul limbajului de programare Java folosind **libraria javax.swing.\***. Interfata grafica creata arata in felul urmator:



Cand se detecteaza un obiect dintr-un anumit interval, cifrele de pe radar vor avea culoarea rosie. In acest timp linia va traversa semicercul pentru a observa mai usor unghiul la care se afla senzorul pe suprafata respectiva

## 4. Testarea propriu-zisa a proiectului

Implementarea si design-ul proiectului se afla in imaginea de mai jos:



Pentru a incepe executia proiectului se va rula programul in java. O data ce programul incepe rulara se va deschide interfata grafica si motorul servo se va afla la unghiul 0. Acesta va incepe "scanarea" suprafetei cu o anumita viteza ce am setat-o in codul C. Atunci cand motorul ajunge la unghiul de 180 de grade, acesta face o pauza de 500ms dupa care incepe sa se roteasca in sens opus repetand procesul. In tot acest timp, senzorul ultrasonic scaneaza suprafata si trimite datele in aplicatia Java care sunt afisate in interfata grafica(unghiul si distanta unui obiect fata de radar). Atunci cand un obiect se afla la o anumita distanta fata de radar, intervalul afisat in interfata grafica isi va schimba culoarea in cea rosie. In tot acest timp, in interfata grafica se misca o linie pentru a putea vizualiza cat mai usor pozitia senzorului ultrasonic si directia in care acesta scaneaza suprafata.





## **5. Concluzii și dezvoltări ulterioare**

În concluzie, dezvoltarea unui asemenea proiect a făcut mult mai ușoară înțelegerea folosirii unei plăcuțe Arduino și componentele acesteia în diferite proiecte care pot fi mult mai complexe. Pe lângă acest lucru, a avut și scopul de a aprofunda cunoștințele legate de limbajul de programare Java. Mai mult de atât, am putut învăța cum sunt transmise datele la calculator de la componentele externe prin portul serial și cum acestea pot fi citite în diferite limbaje de programare.