Proiect SI 2019

Zamfir Mihai-Adrian

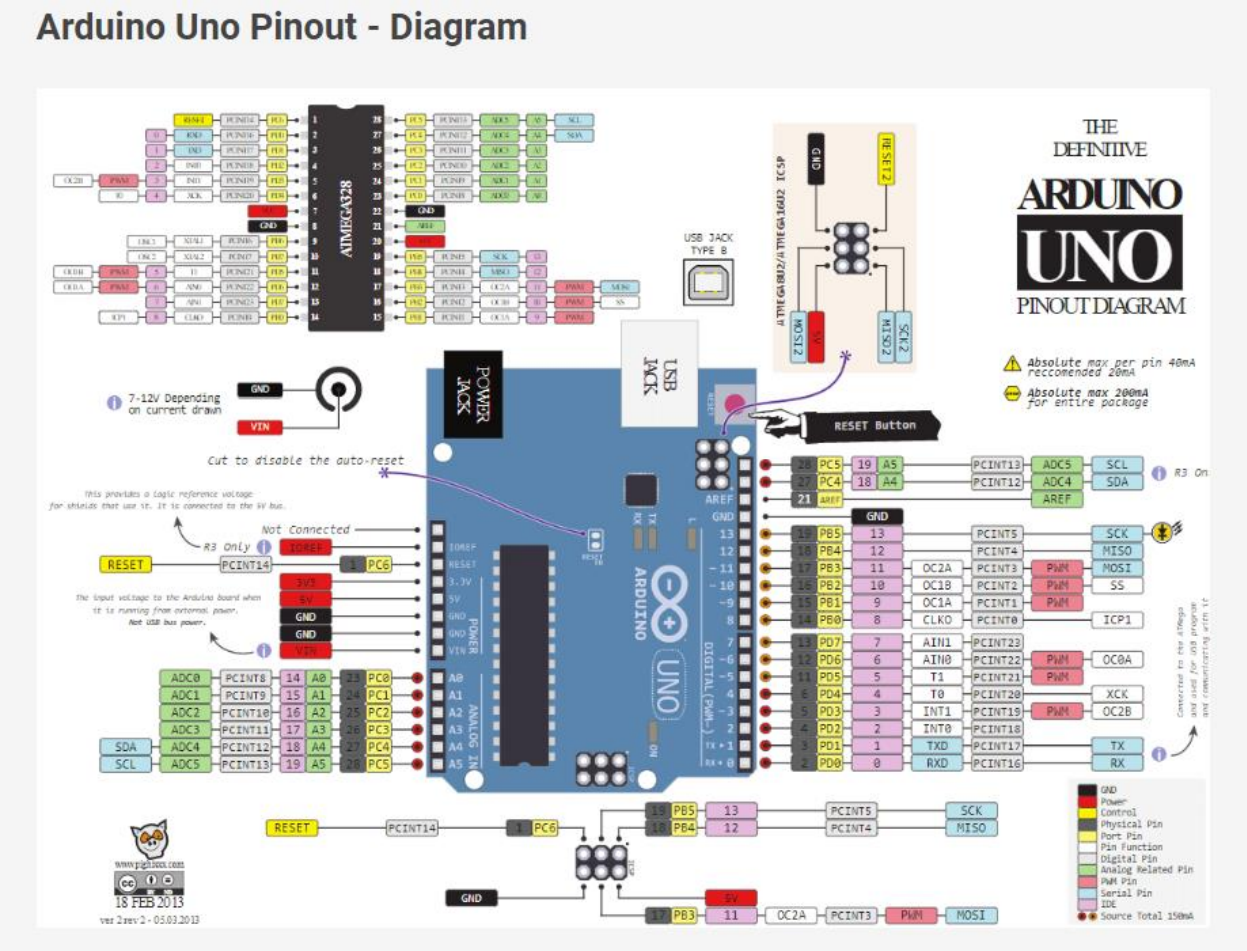
Sgr.6.3, CTI anul III

Enunt

Realizarea unui proiect cu scopul controlarii cursorului prin intermediul senzorilor de distanta hc-sr04 si microcontroller-ului Arduino Uno.

Microcontroller-ul folosit

Microcontroller-ul folosit in acest proiect este Arduino Uno.



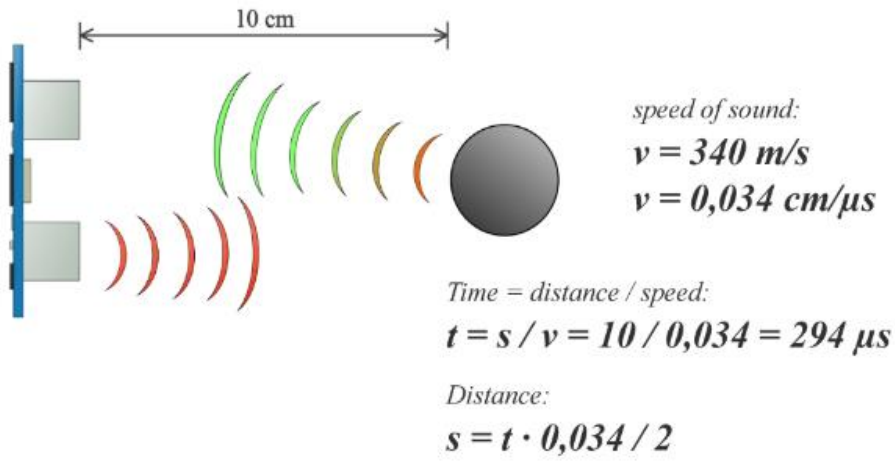
Schema Pinilor unui Arduino Uno

Am ales Arduino Uno deoarece el pune la dispozitie prelucrarea de semnale analog care vin dinspre senzorii cu care comunica (A se vedea Schema Montajului). Pinii folositi sunt 5/6 si 9/10 .

In cadrul proiectului s-a folosit sistemul de timer (pentru delay de durate controlabile) si interfata USB .

Functionarea Proiectului

Proiectul se bazeaza pe senzorii HC-SR04 .



Acestia trimit o unda sonora , aceasta loveste un obiect si unda se reflecta inapoi spre senzor. Stiind viteza sunetului si timpul de parcurgere (de 2 ori , ping & rebound) , putem afla foarte usor distanta obiectului relativ la senzorul nostru.

Acest calcul va fi implementat pe Arduino si distanta va fi trimisa prin interfata seriala catre aplicatia PC care va manipula cursorul mouse-ului. Manipularea aceasta se realizeaza pe doua axe oX si oY. Avand doi senzori , fiecare manipuleaza o axa.

Senzorul HC-SR04

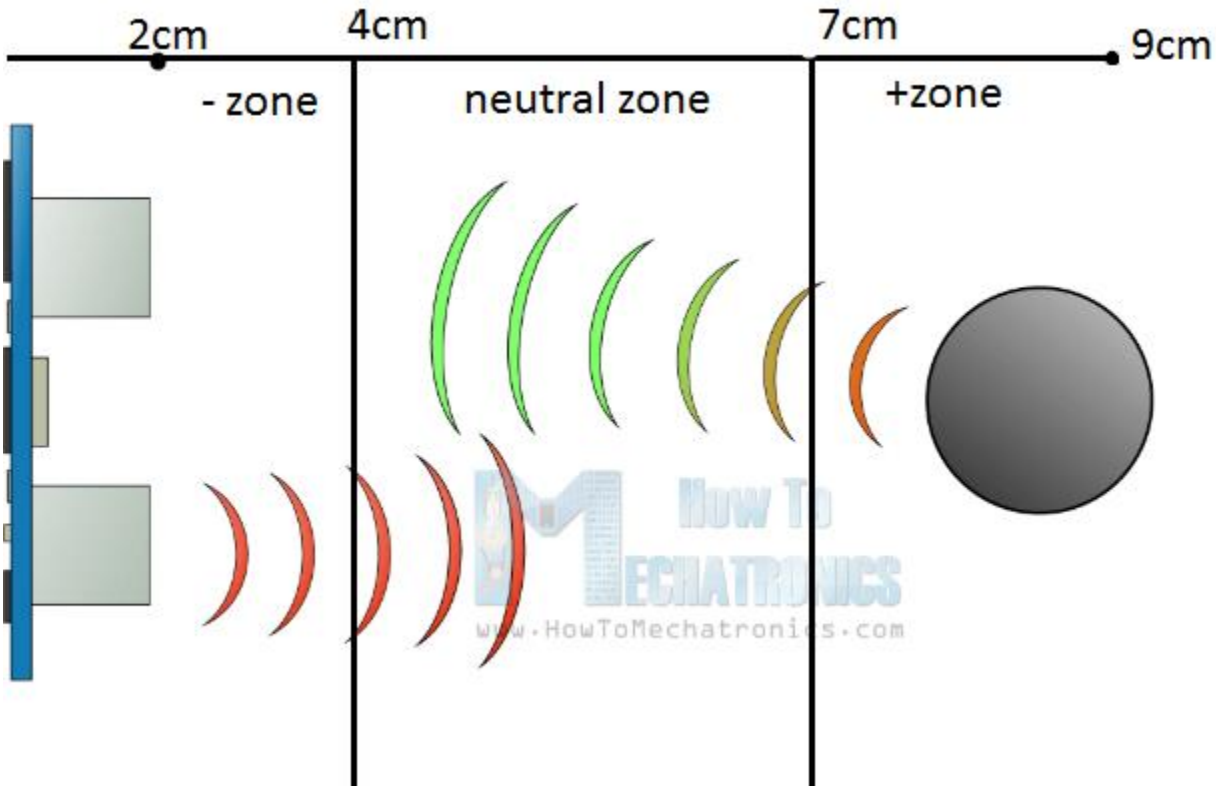
Ultrasonic Sensor Pin Configuration

Pin Number	Pin Name	Description
1	Vcc	Pin alimentare . De obicei 5v.
2	Trigger	Pin de input . Trebuie mentinut 10us pentru a trimite ultrasonetul sia-l receptiona
3	Echo	Pin de output . Aici se receptioneaza ultrasunetul trimis de Trigger
4	Ground	Pin de conectat la ground-ul sistemului.

HC-SR04 Sensor Features

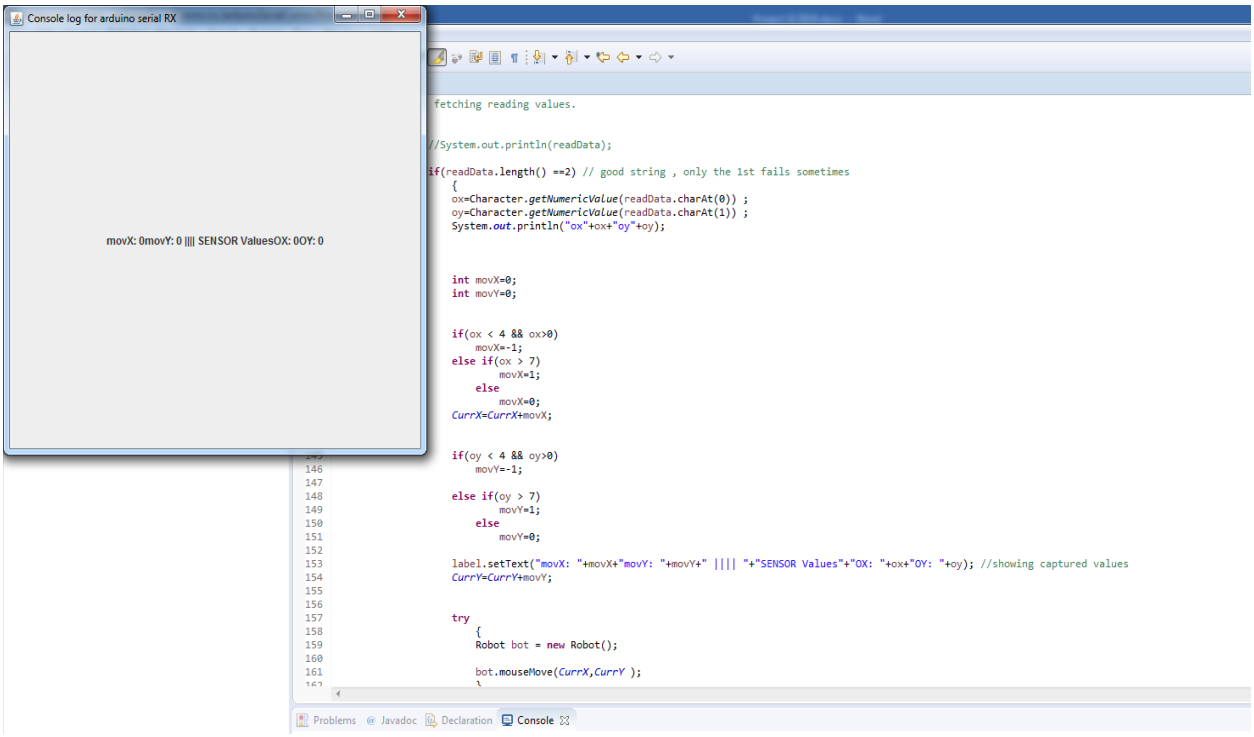
- Tensiune de operare: +5V
- Interval de masurare (teoretic): 2cm - 450cm
- Interval de masurare(practic): 2cm - 80cm
- Precizie: 3mm
- Unghi de masurare: <15°
- Curent de operare: <15mA
- Frecventa de operare: 40Hz

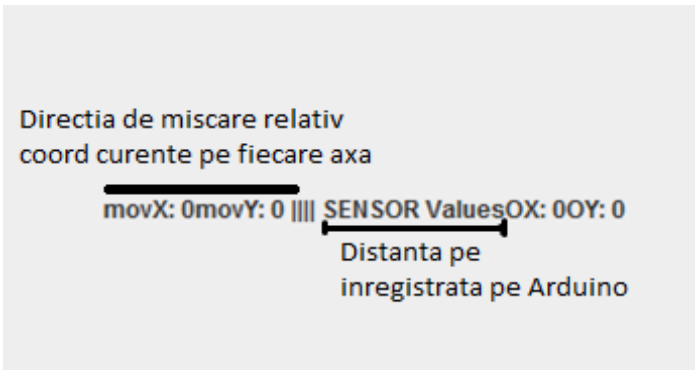
Manipularea Cursorului



In functie de distanta fata de senzor , definim 3 zone. -Zone este zona in care coordonata cursorului scade (pe axa corespunzatoare). +Zone creste coordonata respectiva , iar zona neutral face ca cursorul sa ramana in acelasi loc.

Vizualizare a aplicatiei





Aplicatia preia distantele calculate si trimise de Arduino pe interfata seriala. De asemenea afiseaza pe un Jframe datele prelucrate. In timpul in care aplicatia se executa , mouse-ul este controlat de ea , mai bine zis de datele pe care le preia de pe interfata seriala.

Programare Arduino

```
int trigPin = 9;
int echoPin = 10;
int trigPin2 = 5;
int echoPin2 = 6;



void setup() {
  Serial.begin (9600); // baud rate 9600
  pinMode(trigPin, OUTPUT); // configurare pini , ca in documentatie
  pinMode(echoPin, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  }

void loop() {
  int duration, distance;
  digitalWrite (trigPin, HIGH);
  delayMicroseconds (10);
  digitalWrite (trigPin, LOW);
  duration = pulseIn (echoPin, HIGH);
  distance = (duration/2) / 29.1; // aplicarea formulei , dupa simplificari
    // Serial.print("");
     if(distance>9) // filtrare rezultate reziduale
                    Serial.print("0");
           else
```

```
        Serial.print(distance);
    delay(1); // delay-ul este pentru stabilitate intre writing-uri. 1ms


   int duration2, distance2;
   digitalWrite (trigPin2, HIGH);
   delayMicroseconds (10);
   digitalWrite (trigPin2, LOW);
   duration = pulseIn (echoPin2, HIGH);
   distance2 = (duration/2) / 29.1; // aplicare formula , dupa simplificari
   Serial.print("");
   if(distance2>9)  // aplicare formula , dupa simplificari
    Serial.print("0");
   else
    Serial.print(distance2);

    Serial.println();
    delay(1); // delay-ul este pentru stabilitate intre writing-uri. 1ms




  }
// Code ends here.
```

Programare Aplicatie in Java

```
package arduinoSerialComm;

import java.awt.AWTException;
import java.util.Scanner;
import javax.swing.JFrame;
import javax.swing.JSlider;
import com.fazecast.jSerialComm.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;

import java.awt.AWTException;
import java.awt.BorderLayout;
import java.awt.Component;
import java.awt.Frame;
import java.awt.Robot;
import java.awt.TextArea;
import java.awt.event.InputEvent;
import java.awt.event.KeyEvent;

public class Main {


	public  static int CurrX=500; // coord start
	public static int CurrY=500; // coord start




	public static void main(String[] args) {

		int ox=0;;
		int oy=0;;
```

```
            int stop=0;


            String message;
            JFrame window = new JFrame();
            JSlider slider = new JSlider();
            window.setTitle("Console log for arduino serial RX");
            window.setSize(500,500);
            //slider.setMaximum(20);
            //window.add(slider);


            JLabel label = new JLabel("Opened app
succesfully",JLabel.CENTER); // configurare pentru afisari
            label.setAlignmentX(0);
            label.setAlignmentY(0);
            window.setVisible(true);
            window.add(label);


      window.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
            //ca idee de baza , daca nu aveam aceasta linie aplicatia nu
se inchidea , insemnand ca si citirea de pe interfata seriala se
executa continuu , si in background.

            // cand fereastra este inchisa , si aplicatia se inchide ,
astfel si comunicarea se inchide .


            //window.pack();
            window.setVisible(true);

            SerialPort[] ports = SerialPort.getCommPorts();
            System.out.println("Select a port:");
            int i = 1;
            for(SerialPort port : ports)     // creeare legatura de
comunicare ,sloturile COM se scaneaza , lasand la alegerea
utilizatorului la care dintre ele este coenctat microcontolerul)
                  {label.setText(i +  ": " + port.getSystemPortName());
                  System.out.println(i++ +  ": " +
port.getSystemPortName());
                  }
            Scanner s = new Scanner(System.in);
            int chosenPort ;
            String response=JOptionPane.showInputDialog("Choose port");
```

```
            chosenPort=Integer.parseInt(response);


            SerialPort serialPort = ports[chosenPort - 1];

            if(serialPort.openPort())
                        {System.out.println("Port opened successfully.");
                        label.setText("Port opened successfully.");
                        }
                  else{
                        System.out.println("Unable to open the port.");
                        label.setText("Unable to open the port.");
                  return;}


            //serialPort.setComPortParameters(9600, 8, 1,
SerialPort.NO_PARITY);

      serialPort.setComPortTimeouts(SerialPort.TIMEOUT_READ_SEMI_BLOCKI
NG, 0, 0); // citire de pe interfata seriala ,continuu pana cand
aplicatia iese sau numai este linie de citit

            Scanner data = new Scanner(serialPort.getInputStream());

            String readData="";



            while(data.hasNextLine() ){
                  try
                  {     //System.out.println(readData );
                        readData=data.nextLine();

                        //System.out.println(readData);


                  }
                  catch(Exception e){}
```

```
                // formatul este  xy , unde x este un digit  pe axa OX
iar y este pe axa OY


                { // prindem valorile


                     //System.out.println(readData);

                     if(readData.length() ==2) // good string , only
the 1st fails sometimes
                          {

      ox=Character.getNumericValue(readData.charAt(0)) ;

      oy=Character.getNumericValue(readData.charAt(1)) ;
                          System.out.println("ox"+ox+"oy"+oy);



                          int movX=0;
                          int movY=0;


                          if(ox < 4 && ox>0)
                               movX=-1;
                          else if(ox > 7)
                                    movX=1;
                               else
                                    movX=0;
                          CurrX=CurrX+movX; // calculam /actualizam
coord cursor


                          if(oy < 4 && oy>0)
                               movY=-1;

                          else if(oy > 7)
                                    movY=1;
                               else
                                    movY=0;  // calculam /actualizam
coord cursor

                          label.setText("movX: "+movX+"movY: "+movY+"
|||| "+"SENSOR Values"+"OX: "+ox+"OY: "+oy); //showing captured values
                          CurrY=CurrY+movY;
```

```
                              try
                              {
                                    Robot bot = new Robot(); //clasa care
se ocupa de miscarea cursorului la coordonatele dorite

                                    bot.mouseMove(CurrX,CurrY );
                              }
                              catch (AWTException e)
                              {
                                    e.printStackTrace();
                              }



                              }// end valid message if



                        }
                  }
                  System.out.println("Done.");
                  return;
            }

}
```

Bibliografie

1. http://howtomechatronics.com
2. https://fazecast.github.io/jSerialComm/ (JAR that provides the API for accessing the serial interface)
3. https://components101.com/ultrasonic-sensor-working-pinout-datasheet
4. https://www.datasheetspdf.com/pdf/1291829/Cytron/HC-SR04/1

Acces resurse : https://github.com/zamfir-m-a/PROIECT-SI-2019