DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING
College of Engineering and Technology
SRM Institute of Science and Technology

MINI PROJECT REPORT

ODD Semester, 2023-2024

Lab code & Sub Name	:	(18ECO108J) EMBEDDED SYSTEM DESIGN USING ARDUINO
Year & Semester	:	3rd Year,5th Semester
Project Title	:	Car Parking Sensor with Arduino and Distance Sound Effects
Lab Supervisor	:	**Dr.V. Padmajothi** AP/ECE
Team Members	:	1.Swetha Suresh (Reg.No:RA2111026010259) 2.Jephrin Esther (Reg.No:RA2111026010215) 3.Lohith Kumar(Reg.No:RA2111026010232)

Particulars	**Max. Marks**	**Marks Obtained**
		Name: Swetha Suresh
		RegisterNo: RA2111026010259
Program and Execution	20	
Demo verification &viva	15	
Project Report	05	
Total	**40**	

Date :

Staff Name :

Signature :

Car Parking Sensor with Arduino and Distance Sound Effects

OBJECTIVE:

The main objective is to enable accurate and convenient car parking through the use of Arduino-based sensors, accompanied by distance-based sound effects for enhanced spatial awareness.

ABSTRACT:

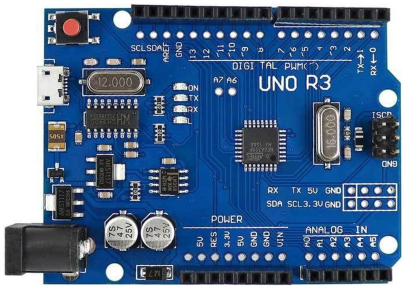
In our increasingly crowded urban environments, finding a suitable parking spot can be a daily challenge for motorists. Whether it's a tight parallel parking space on a busy city street or a crowded parking lot at a shopping mall, the process of parking can be stressful and time-consuming. This is where the "Car Parking Sensor with Arduino and Distance Sound Effects" project comes into play.
The Car Parking Sensor project aims to provide a practical and innovative solution to assist drivers in parking their vehicles safely and with confidence.
By combining Arduino technology with distance measurement sensors and sound effects, this project seeks to make parking easier, more efficient, and less stressful for drivers of all experience levels.

INTRODUCTION:

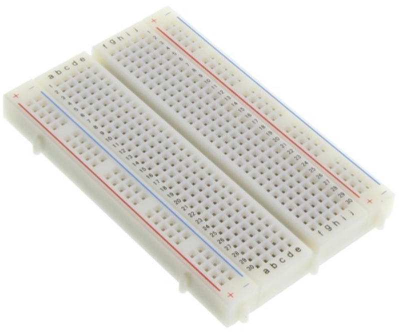
The primary aim of the "Car Parking Sensor with Arduino and Distance Sound Effects" project is to create a smart parking assistance system that offers the benefits

HARDWARE/SOFTWARE REQUIREMENTS:

Arduino (e.g., Arduino Uno): The central brain of the project, responsible for processing sensor data and controlling the LED indicator.



Breadboard: Used for easy and temporary electronic connections, facilitating the assembly of the circuit.



Ultrasonic Sensor (HC-SR04): This sensor is at the heart of the system, allowing the measurement of distances using ultrasonic waves.

Buzzer: Buzzer for Sound

Jumper Wires (Male-to-Male and Male-to-Female):Necessary for connecting the components and creating a functional circuit.

CONCEPTS/WORKING PRINCIPLE

- Buzzer (buzPin) Connection:The variable buzPin is set to 7, indicating that the buzzer is connected to digital pin 7 on the Arduino.

- Ultrasonic Sensor Trigger (trigPin) and Echo (echoPin) Connections:The variable trigPin is set to 8, indicating that the trigger pin of the ultrasonic sensor is connected to digital pin 8 on the Arduino.The variable echoPin is set to 4, indicating that the echo pin of the ultrasonic sensor is connected to digital pin 4 on the Arduino.

- Speed of Sound (speed) Declaration:The variable speed is set to 0.0347, representing the speed of sound in air at room temperature. This is used in the calculation of distance.

- Variables for Buzzing Times (buzNear, buzHigh, buzMid, buzFar, delayFar):These variables are used to set different durations for buzzing the buzzer based on the distance measured. For example, buzNear is set to 20, indicating a short buzzing time for very close proximity, while buzFar is set to 600 for a longer buzzing time for far-off objects. delayFar is also used for a delay after a far object is detected.

- setup() Function:The setup() function configures the pins for the buzzer, trigger, and echo as either output or input as required. The buzzer and trigger pins are set as output, while the echo pin is set as an input.
- loop() Function:The loop() function is the main program loop that continuously measures the distance using the ultrasonic sensor and triggers the buzzer based on the distance value. It generates an ultrasonic pulse, calculates the distance based on the pulse's return time, and then triggers the buzzer with different buzzing times depending on the distance measured.

METHODOLOGY AND PROGRAMS:

- In many of our projects, we need to sense our environment, to know if there is an obstacle in front of us and to know the distance between sensor and obstacle, HC-SR04 ultrasonic distance sensor helps us to solve those problems.
- The HC-SR04 ultrasonic sensor is a low cost distance sensor, widely used in robotics, it uses ultrasound transducers to detect objects.
- When is sensor is powered on an ultrasonic sound wave(at higher wavelength, humans cannot hear) is emitted through one of its transducers, and waiting for the sound to bounce off at some present object, the echo is captured by the second transducer. The distance is proportional to the time it takes for the echo to arrive.
- In Simple terms: The operation is very simple. The sensor sends an ultrasonic sound wave through the trigger or trigger, bounces off the object and the receiver or echo detects the wave. By measuring the time taken for the wave from emission till it gets back we can know the distance.
- The distance is calculated with the below formula which we learnt at school

- **velocity formula:**

- Velocity(v) = Distance(s)/time(t). ⇒ Distance(s)= Velocity(v) X time(t).
- As we know the speed of the sound in air is 343 meters per second(1234.8kmph) at 20°C, and we can measure the time between the emitter and receiver. By multiplying speed with time we can calculate the distance. Which can be done through program code and processed by microcontrollers. The speed increases or decreases 0.6 m/s per degree centigrade. We can be more accurate if we use a temperature sensor like the LM35 .

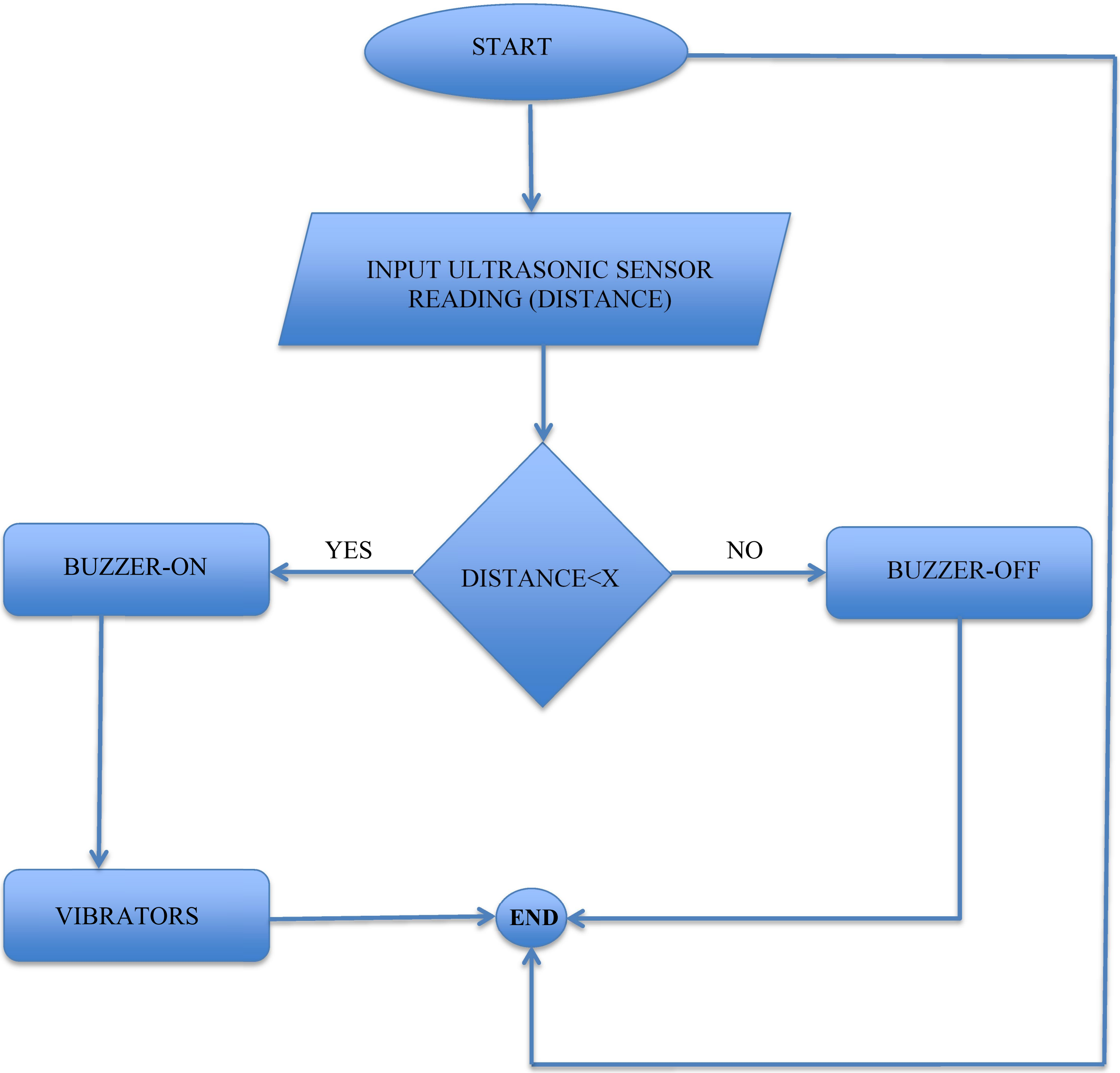
Source Code:

```
// Arduino parking sensor code www.circuitschools.com
```

```
int buzPin = 7;       //declare pin for Buzzer;
int trigPin = 8;      //declare pin for trigger pin of UltraSonic sensor;
int echoPin = 4;      //declare pin for echo pin of UltraSonic sensor;
float speed = 0.0347; //declare speed of sound in air @ room temp;
float dist;           //declare variable for containing distance sensed;
float pingTime;       //declare variable for containing echo time;
int buzNear = 20;     //declare buzzing time for very close proximity;
int buzHigh = 50;     //declare buzzing time for close proximity;
int buzMid  =130;     //declare buzzing time for mid proximity;
int buzFar = 600;     //declare buzzing time for far off object;
int delayFar = 260;


void setup() {
  Serial.begin(9600);
  pinMode(buzPin,OUTPUT);     //set buzzer & trigger pin as outpin;
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);     //set echo pin as input;

}

void loop() {
  digitalWrite(trigPin,LOW);
  delayMicroseconds(20);
  digitalWrite(trigPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin,LOW);          //creating a pulse for sensing
distance;
  pingTime = pulseIn(echoPin,HIGH);   //read the echoTime, &hence
the distance;
  dist = (speed*pingTime*0.5);
  Serial.print("Distance: ");
  Serial.println(dist);

  if(dist<=10.0){
    digitalWrite(buzPin,HIGH);         //simple conditional statements
changing frequency based upon the distance sensed
```

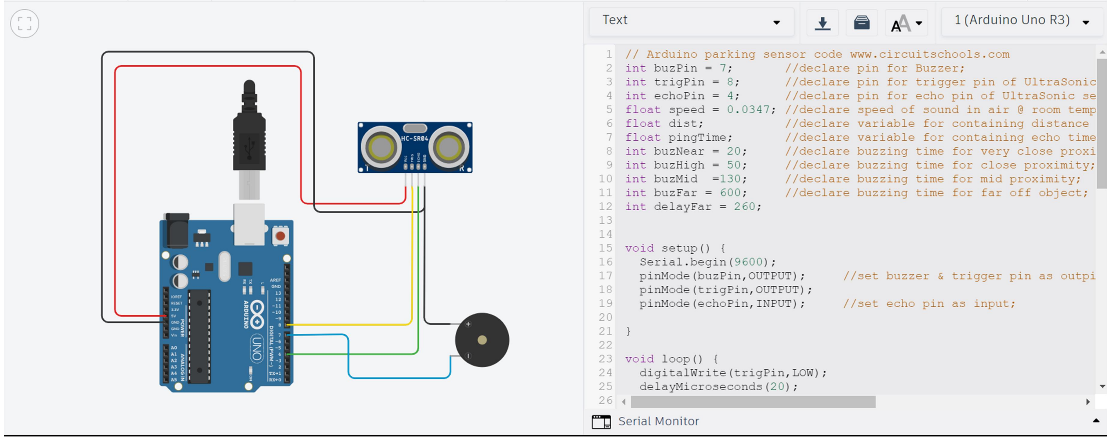
```
    delay(20);
    digitalWrite(buzPin,LOW);
    delay(20);
  }
  else if(dist<=30.0 && dist>10.0)
  {
    digitalWrite(buzPin,HIGH);
    delay(buzHigh);
    digitalWrite(buzPin,LOW);
    delay(buzHigh);
  }
  else if((dist>30.0) && (dist<60.0))
  {
    digitalWrite(buzPin,HIGH);
    delay(buzMid);
    digitalWrite(buzPin,LOW);
    delay(buzMid);
  }
  else if(dist>=60.0 && dist<120.0)
  {
    digitalWrite(buzPin,HIGH);
    delay(buzFar);
    digitalWrite(buzPin,LOW);
    delay(delayFar);
  }

}
```

FLOWCHART:



START
INPUT ULTRASONIC SENSOR READING (DISTANCE)
DISTANCE<X
YES
NO
BUZZER-ON
BUZZER-OFF
VIBRATORS
END

OUTPUT:



```
// Arduino parking sensor code www.circuitschools.com
int buzPin = 7;        //declare pin for Buzzer;
int trigPin = 8;       //declare pin for trigger pin of UltraSonic
int echoPin = 4;       //declare pin for echo pin of UltraSonic se
float speed = 0.0347; //declare speed of sound in air @ room temp
float dist;            //declare variable for containing distance
float pingTime;        //declare variable for containing echo time
int buzNear = 20;      //declare buzzing time for very close proxi
int buzHigh = 50;      //declare buzzing time for close proximity;
int buzMid  =130;      //declare buzzing time for mid proximity;
int buzFar = 600;      //declare buzzing time for far off object;
int delayFar = 260;


void setup() {
  Serial.begin(9600);
  pinMode(buzPin,OUTPUT);     //set buzzer & trigger pin as outpi
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);     //set echo pin as input;

}

void loop() {
  digitalWrite(trigPin,LOW);
  delayMicroseconds(20);
```

CONCLUSIONS:

The "Car Parking Sensor with Arduino and Distance Sound Effects" project offers an innovative solution to enhance the parking experience for motorists. Through the integration of an Arduino Uno microcontroller, an HC-SR04 ultrasonic distance sensor, and a buzzer for sound feedback, this project aims to provide several key advantages.

- **Safety**: Improve safety by alerting drivers to obstacles and potential collisions, reducing the risk of accidents during parking maneuvers.
- **Efficiency**: Streamline the parking process, helping drivers find and occupy parking spaces more quickly, thus reducing traffic congestion and fuel consumption.
- **User-Friendly**: Develop a user-friendly interface that is easy for drivers to understand and operate, making it accessible to individuals of all skill levels.
- **Cost-Effective:** Offer a cost-effective alternative to commercially available parking assist systems, allowing more drivers to benefit from this technology.
- **Customizable**: Allow users to configure the system to suit their vehicle's specific requirements, including the ability to set distance thresholds and choose from various sound effects.
- **Enhanced Parking Experience:** Provide an enhanced parking experience by combining visual and auditory feedback, helping drivers navigate in tight spaces with confidence.

In summary, the "Car Parking Sensor with Arduino and Distance Sound Effects" project successfully addresses the challenges associated with parking in congested environments. By merging technology and creative sound feedback, it makes parking safer and more efficient, reduces stress for drivers, and enhances the overall parking experience. This project exemplifies the power of DIY solutions in making our daily routines more convenient and secure.

REFERENCES:

https://www.watelectronics.com/ultrasonic-sensor/