OBJECT TRACKING SYSTEM

MINI PROJECT REPORT

Submitted by

KARNASSAGAR S	**9517202109026**
MUTHU NITHEESH R	**9517202109036**
SIVA PRASANTH M	**9517202109049**

in

19AD752 – INTELLIGENT SYSTEMS FOR IOT LABORATORY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI

AUTONOMOUS

October 2024

MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI
AUTONOMOUS

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

BONAFIDE CERTIFICATE

This is to certify that it is the bonafide work of **“KARNASSAGAR S (9517202109026), MUTHU NITHEESH R (9517202109036), SIVA PRASANTH M (9517202109049)”** for the Mini project titled **“Object Tracking System”** in 19AD752–Intelligent Systems for IOT Laboratory during the seventh semester June 2024–November 2024 under my supervision.

SIGNATURE	**SIGNATURE**
Dr.A.Shenbagarajan ,	**Dr. J. Angela Jennifa Sujana,**
Associate Professor,	**Professor & Head,**
AI&DS Department,	AI&DS Department,
Mepco Schlenk Engg.College,	Mepco Schlenk Engg.College,
Sivakasi	Sivakasi

ABSTRACT

Our project presents an object tracking system using ultrasonic sensors and servo motors,designed for real-time detection and tracking of objects within a defined area.The system utilizes two ultrasonic sensors positioned along the X and Y axes to measure distances and determine the position of a target. These distance readings are processed by a microcontroller, such as an Arduino, which then adjusts two servos to track the object's movement.The ultrasonic sensors measure the distance to the object by emitting sound waves and detecting the time it takes for the echo to return, allowing for precise distance calculation. The X-axis distance determines the horizontal angle, while the Y-axis distance fine-tunes the vertical adjustment, enabling accurate tracking. The system's servos, positioned on both the left and right, adjust their angles in response to these distance measurements, ensuring that the tracked object remains centered in the field of view.The project aims to provide a cost-effective and scalable solution for automated object tracking, suitable for applications in security systems, robotics, and automated monitoring. With a simple design and use of easily available components, this object tracking system demonstrates the potential of combining ultrasonic distance measurement with servo control for practical real-world applications.

ACKNOWLEDGEMENT

First and foremost we **praise and thank "The Almighty",** the lord of all creations, who by his abundant grace has sustained us and helped us to work on this project successfully.

We really find unique pleasure and immense gratitude in thanking our respected management members**,** who are the backbone of our college.

A deep bouquet of thanks to respected Principal **Dr.S.Arivazhagan M.E.,Ph.D.,** for having provided the facilities required for our mini project.

We sincerely thank our Head of the Department **Dr. J. Angela Jennifa Sujana M.E.,Ph.D.,** Professor & Head, Department of Artificial Intelligence and Data Science, for her guidance and support throughout the mini project .

We extremely thank our project coordinator **Dr.A.Shenbagarajan M.E.,Ph.D,** Associate Professor, Department of Artificial Intelligence and Data Science, who inspired us and supported us throughout the mini project.

We extend our thanks for our project coordinator **Dr.E.Emmerson Nithyaraj M.E.,Ph.D,** Associate Professor, Department of Artificial Intelligence and Data Science, who inspired us and supported us throughout the mini project.

We extend our heartfelt thanks and profound gratitude to all the faculty members of Artificial Intelligence and Data Science department for their kind help during our mini project work.

We also thank our parents and our friends who had been providing us with constant support during the course of the mini project work.

TABLE OF CONTENT

LIST OF FIGURES

1. INTRODUCTION

1.1 INTRODUCTION

Object tracking is a crucial technology in various fields such as security, robotics, automation, and surveillance.Our project focuses on developing an object tracking system using ultrasonic sensors and servo motors, capable of detecting and following the movement of an object in real time. The system employs two ultrasonic sensors for distance measurement along the X and Y axes, allowing it to determine the position of an object within a specified area. By using the distance data from these sensors, the system adjusts the angles of two servos to keep the object centered within its range of detection.The Automatic Light Intensity Controller aims to solve this problem by dynamically adjusting the brightness of vehicle lights based on the distance and height of oncoming vehicles. Using two ultrasonic sensors, this system detects the proximity and size of an approaching vehicle and automatically adjusts the light intensity to an appropriate level. This ensures clear visibility for the driver while minimizing the chances of dazzling other road users. The system is designed to be cost-effective, energy-efficient, and easily adaptable to modern vehicle lighting systems.

1.2 OBJECTIVES

The aim of our project is to track the object using ultrasonic sensor and servo based. To achieve this objective, we consider:

1. **Design and Implement a Real-Time Object Tracking System** to detect and track the position of an object in real-time using ultrasonic sensors and servo motors.
2. **Integrate Distance Measurement Technology** Utilize ultrasonic sensors to accurately measure distances along the X and Y axes, enabling the precise calculation of an object's position.

3. **Automate Servo Motor Control** that adjusts the angles of two servo motors based on the object's position, allowing smooth and continuous tracking.
4. **Provide a Cost-Effective Solution** by creating an object tracking system using readily available and affordable components, ensuring that the system is accessible for a wide range of applications and users.
5. **Ensure System Stability and Accuracy** by optimizing sensor placement and fine-tuning the control logic for stable servo movements.
6. **Demonstrate Potential Applications,** To showcase the system's adaptability for various use cases, such as security surveillance, automated monitoring, and robotic vision systems, highlighting its practical value in real-world scenarios.

1.3 SCOPE OF THE PROJECT

Object tracking system project focuses on the tracking of objects based system. The scope of the project includes the following:

1. **Range of Detection:**
   - The system is designed to detect objects within the range of the ultrasonic sensors, typically up to 2 meters, depending on the sensor specifications.
   - The X and Y axes are covered, allowing tracking in a two-dimensional space.
2. **Hardware Components:**
   - The system is built using cost-effective and commonly available hardware components, including a microcontroller (like Arduino), two ultrasonic sensors, and two servo motors.
   - The hardware design focuses on simplicity, making it suitable for hobbyists, students, and small-scale automation projects.

3. **Adaptability for Different Applications:**
    - While the primary scope involves object tracking, the system is adaptable for various practical applications, such as automatic surveillance, robotic vision systems, and motion-sensitive devices.
    - Its flexibility allows for adjustments to enhance its range, accuracy, and response time as needed.
4. **Educational and Learning Value:**
    - Our project serves as an educational tool for understanding the basics of distance measurement using ultrasonic sensors, microcontroller programming, and servo motor control.
    - It is well-suited for students and hobbyists looking to gain hands-on experience with IOT and automation concepts.
5. **Scalability and Future Enhancements:**
    - The system has the potential for further enhancements, such as integrating additional sensors for 3D tracking, incorporating wireless communication for remote monitoring, or adding a camera for visual feedback.
    - These enhancements could expand its usability in more complex automation systems.

2. PROPOSED SOLUTION

2.1 BLOCK DIAGRAM

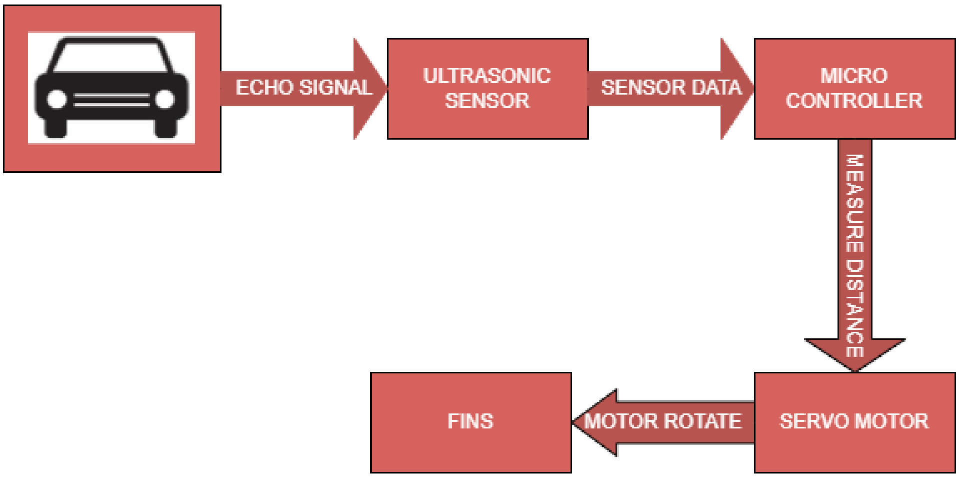

Figure 2.1.1 Block Diagram

2.2 SYSTEM BEHAVIOR:

1. Distance-Based Control:

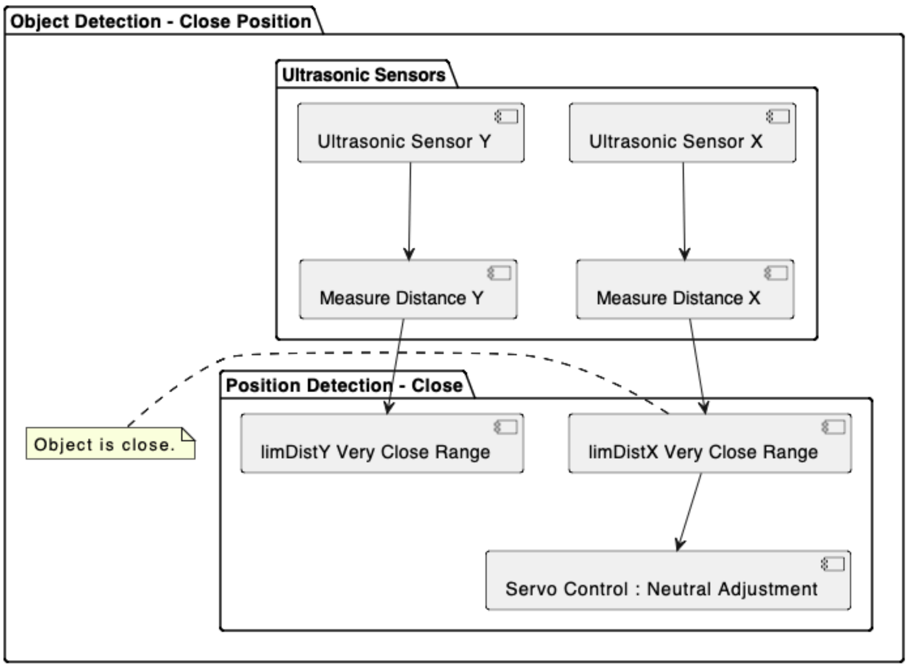

Fig 2.2.1 Case 1: Object detected - Close Position (limDistX >= 4 cm and limDistY >= 11 cm)

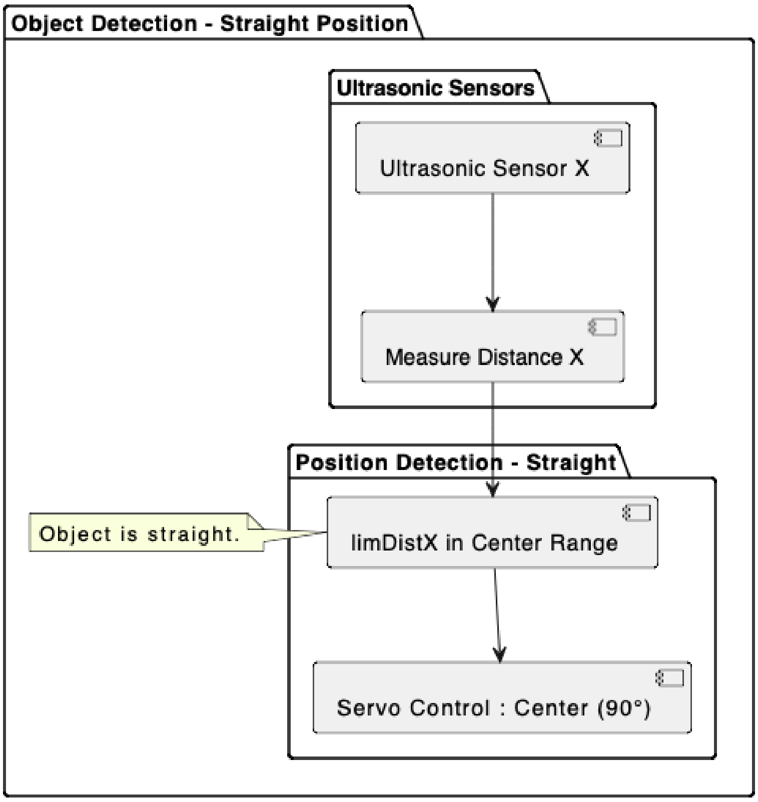

Fig 2.2.2 Case 2: Object detected - Straight Position (36 cm >= limDistX <= 4 cm)

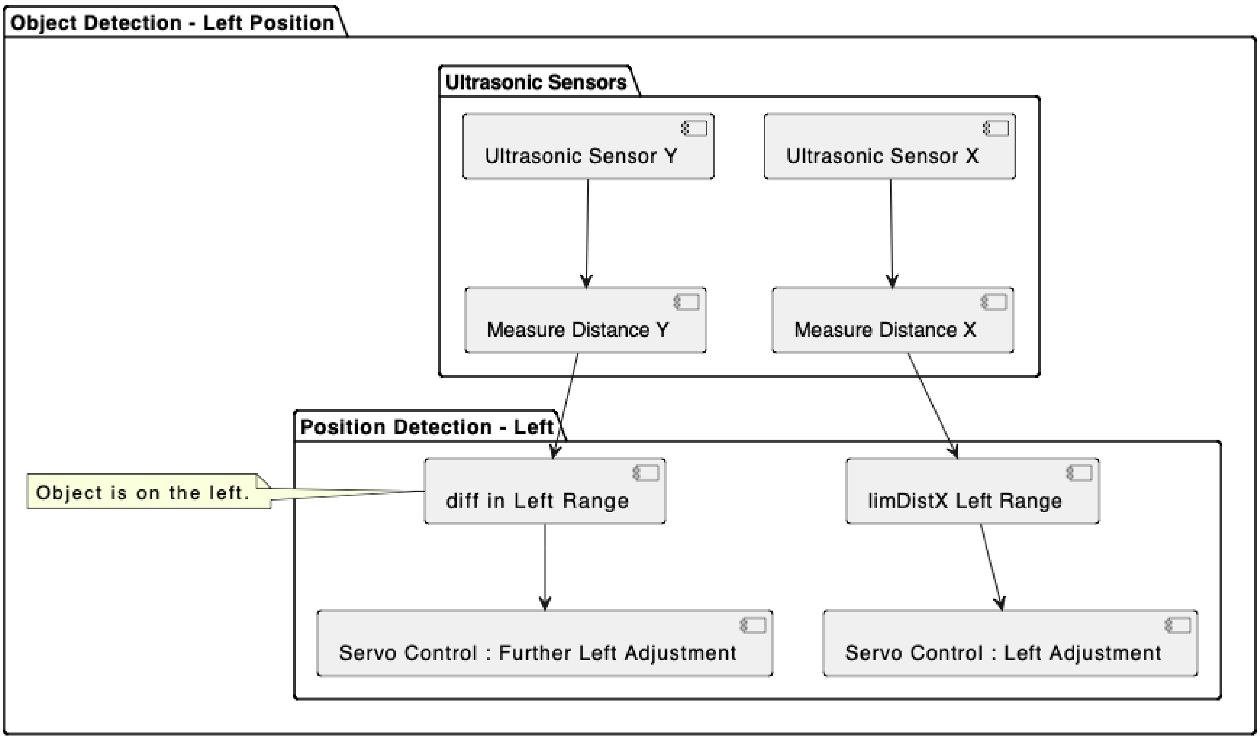

Fig 2.2.3 Case 3: Object detected - Left Position (limDistX >= 4cm and diff > 20 cm)

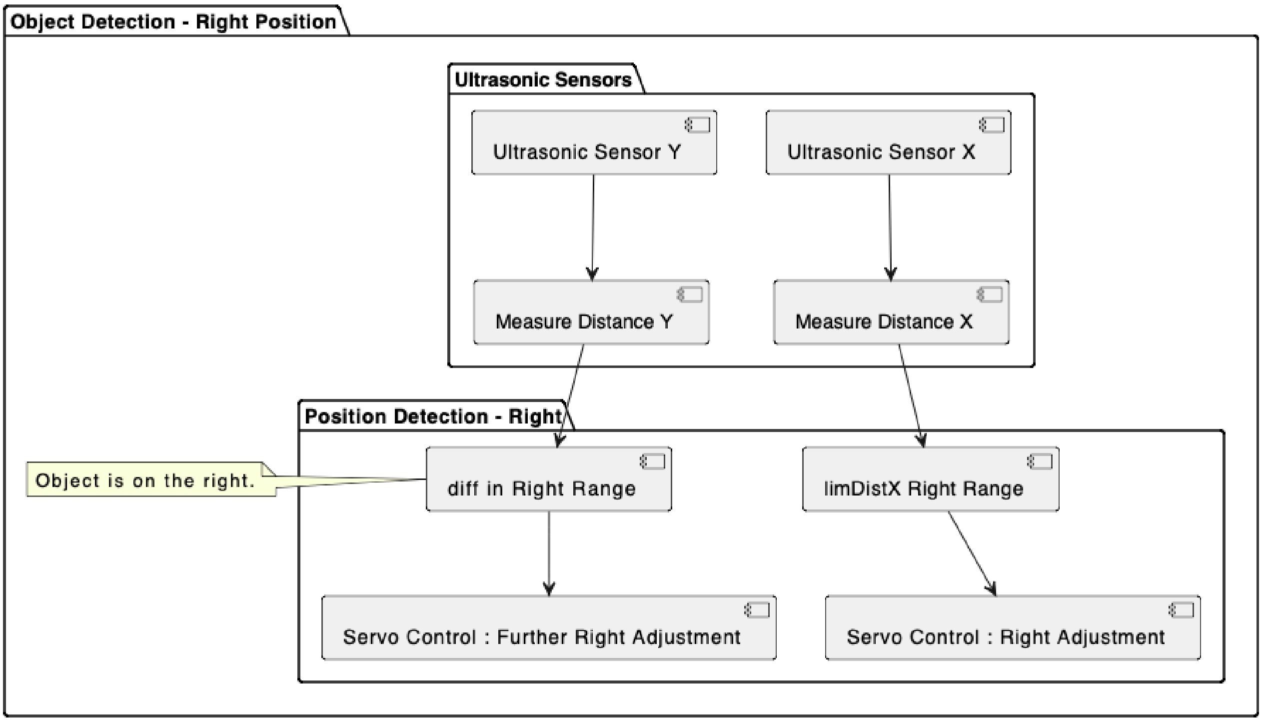

Fig 2.2.1 Case 4: Object detected - Right Position (limDistX >= 36 cm and diff < 20 cm)

2. **Servo Control:**

- **If the object distance is within the threshold**, the servo motor moves to a calculated angle between **45° and 135°**.
- **If the height of the object exceeds a threshold**, the motor adjusts the angle to track the object with a reduced intensity or slower movement.
- **The servo angle is mapped proportionally** to the object's distance and height, ensuring smooth tracking.

2.3 CIRCUIT DIAGRAM

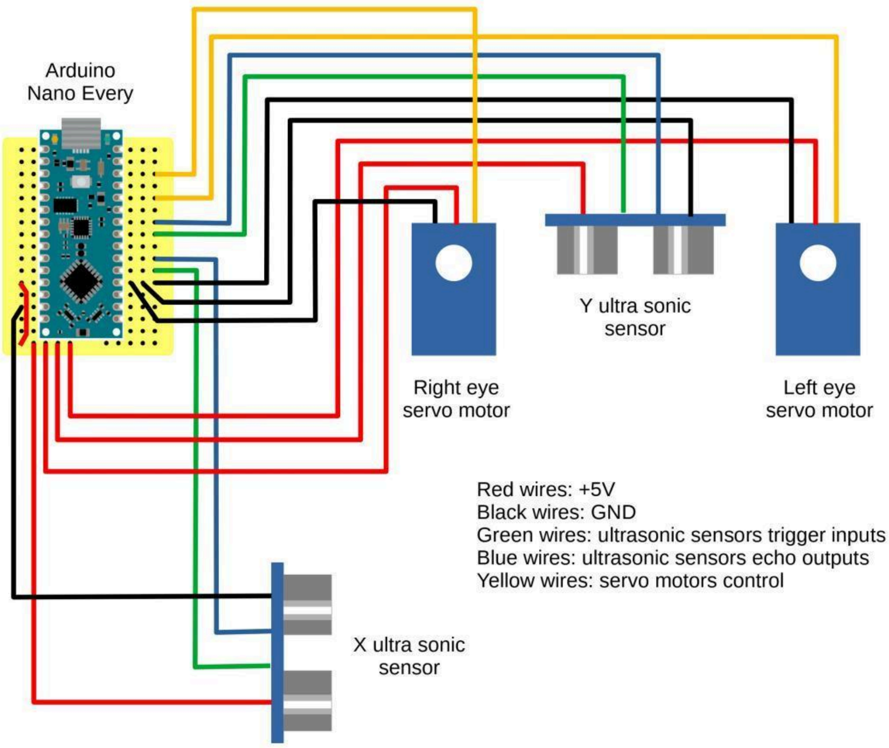

Figure 2.3.1 Circuit Diagram

2.4 HARDWARE REQUIREMENTS

- Arduino NANO Every
- Jumper wires
- USB Cable
- Servo Motor
- Ultrasonic Sensor
- Breadboard

2.5 SOFTWARE REQUIREMENT

- Arduino IDE Software

2.6 ARDUINO NANO EVERY

- The Arduino NANO Every is a compact, affordable microcontroller board designed for simple and flexible projects.
- It features the ATmega4809 microcontroller, offering 48KB Flash, 6KB SRAM, and 256 bytes of EEPROM.
- It has 14 digital I/O pins, 8 analog input pins, and operates at 5V. The board includes PWM outputs, a micro-USB port for programming, and is suitable for small-scale IoT applications, robotics, and automation projects.
- Its compact size makes it ideal for breadboard-based prototyping.

2.6.1 ARDUINO NANO EVERY DIAGRAM

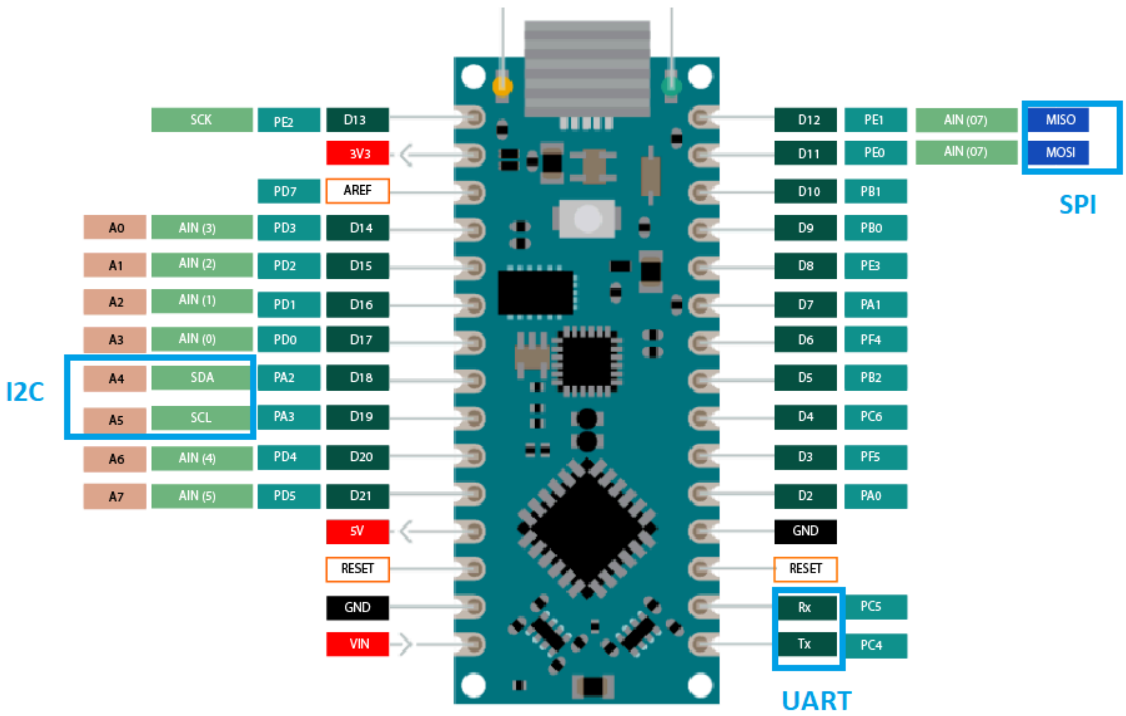

Figure 2.6.1 Arduino NANO Every Board

- **ATmega4809 Microcontroller**: It has an 8-bit AVR processor with 48 KB Flash, 6 KB SRAM, and 256 Bytes EEPROM.
- **ICSP Pins**: Used for programming the microcontroller via an external programmer.
- **Power LED Indicator**: Indicates that the board is powered.
- **Digital I/O Pins**: 14 digital pins (D0–D13) for digital input/output (HIGH/LOW).
- **TX and RX LEDs**: Indicate data transmission (TX) and reception (RX) during

serial communication.

- **AREF Pin**: Used for external analog reference voltage for the analog inputs.
- **Reset Button**: Manually resets the microcontroller.
- **USB**: Micro-USB port for programming and serial communication with a PC.
- **Crystal Oscillator**: A 16 MHz oscillator provides the clock for the microcontroller.
- **Voltage Regulator**: Regulates input voltage to 5V.
- **GND**: Ground pins.
- **Vin**: Input voltage pin for external power.
- **Analog Pins**: 8 analog input pins (A0–A7) with 10-bit resolution for analog-to-digital conversion.

2.7 ULTRASONIC SENSOR

- An ultrasonic sensor operates using sound waves at frequencies above the human hearing range (typically between 20 kHz and 40 kHz).
- These sensors measure the time it takes for the sound waves to travel to a target object and bounce back, making them ideal for distance and proximity sensing.
- The sensor measures the time interval between the moment the sound wave is emitted and when the echo is received. The greater the time difference, the farther the object is from the sensor.
- Ultrasonic sensors have a specific detection cone or field of view, typically in the range of 15° to 30°. This means objects outside of this angle may not reflect sound waves effectively, limiting detection to a narrower field compared to some other sensors (like infrared).
- They provide accurate distance measurements, even for transparent or shiny objects that would cause issues with optical sensors.
- Since the speed of sound varies with environmental conditions, the accuracy of the sensor can be affected by changes in temperature and humidity. Many sensors have built-in temperature compensation to counteract this effect.

TYPICAL SPECIFICATIONS:

- **Range**: Typically, 2 cm to 400 cm, depending on the ultrasonic sensor.
- **Accuracy**: ±1% or better for standard ultrasonic sensors like HC-SR04.
- **Resolution**: 1 mm (based on the ultrasonic sensor's precision).
- **Operating** Frequency: Around 40 kHz for common ultrasonic sensors.
- **Field of View**: Usually between 15° and 30° for standard ultrasonic sensors.
- **Servo Angle**: 0° to 180° for servo motors used in the tracking system.

2.7.1 ULTRASONIC SENSOR DIAGRAM

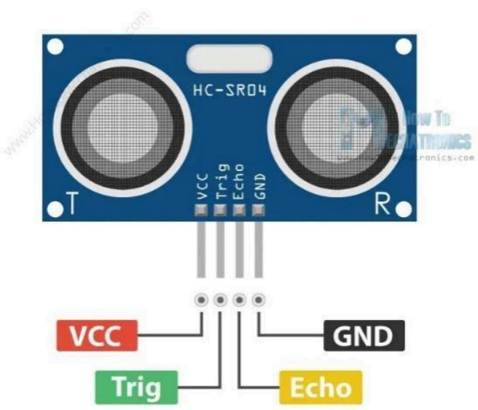

Figure 2.7.1 Ultrasonic Sensor Diagram

1. **Transmitter (Trig):**

   The transmitter is a piezoelectric crystal that generates ultrasonic waves by vibrating at a specific frequency, often around 40 kHz. This vibration generates high-frequency sound waves that propagate through the air.

2. **Receiver (Echo):**

   Like the transmitter, the receiver is also a piezoelectric crystal but is used to detect returning sound waves (echoes). The receiver converts the mechanical vibrations caused by the returning sound waves back into electrical signals.

3. **Distance Calculation:**

Once the ultrasonic wave is emitted, it travels toward the object. When the wave hits an object, it reflects to the sensor. The total travel time of the wave is recorded. Given the speed of sound, the distance to the object is calculated by dividing the travel time in half and multiplying by the speed of sound:

$$Distance = \frac{Speed\ of\ sound * Time\ of\ travel}{2}$$

2.8 SERVO MOTOR

- **Precise Control**: Servo motors provide precise angular positioning, making them ideal for controlled movements.
- **Torque**: These motors deliver sufficient torque for light to medium loads in applications like object tracking.
- **Speed**: Typical speed ranges from 0.1 to 0.2 seconds per 60°.
- **Operating Voltage**: Usually operates between 4.8V and 6V.
- **Angle Range**: 0° to 180° for standard servos.

1. Advantages of Servo Motor

1.1 **Precise Positioning**

Provides accurate angular control, making it ideal for robotics and tracking systems.

1.2 **Low Power Consumption**

Consumes power only when adjusting position, reducing energy use.

1.3 **Compact Size**

Small form factor makes it suitable for integration in lightweight or portable projects like object tracking.

1.4. **Durability**

Solid Construction: With strong materials and reliable electronics, servos can withstand moderate mechanical stress and vibrations.

1.5. **Versatility**

Multi-purpose: Widely used in various applications such as robotic arms, cameras, and control systems.

2.8.1 SERVO MOTOR DIAGRAM

Figure 2.8.1 Micro Servo Motor SG90 Diagram

1. WORKING of Servo Motor

- **Pulse Width Modulation (PWM)**: The servo motor operates using a control signal in the form of PWM. The position of the motor shaft is determined by the duration of the pulse sent to the signal wire.

1.1. Control by Duty Cycle

- **Duty Cycle**: The angle of the motor is controlled by varying the duty cycle of the PWM signal. For example, a 1 ms pulse might move the servo to 0°, a 1.5 ms pulse moves it to 90°, and a 2 ms pulse moves it to 180°.

1.2. Construction of Servo Motor

- **Internal Components**: A servo motor contains a DC motor, a control circuit, and a potentiometer. The potentiometer provides feedback, enabling precise angular positioning by adjusting the motor according to the input signal.

2.9 ARDUINO SOFTWARE

1. Main Components

- Menu Bar: Located at the top, this includes options for File (new, open, save), Edit (cut, copy, paste), Sketch (verify, upload), Tools (board selection, port selection), and Help (documentation, examples).
- Toolbar: This provides quick access to frequently used actions such as verifying code, uploading to the board, and opening the serial monitor.
- Code Editor: The central area where users write their Arduino sketches (programs). It supports syntax highlighting, which helps differentiate between keywords, variables, functions, and comments for better readability.

2. Code Editor Features

- Line Numbers: Displayed on the left side of the editor to help keep track of code lines.
- Auto-Completion: Offers suggestions as you type, helping speed up coding and reducing errors.
- Error Highlighting: Compiles code and highlights any syntax errors or issues, providing feedback before uploading.

3. Output and Serial Monitor

- Output Window: Located at the bottom, this shows messages related to the compilation process, including errors and warnings. It provides feedback after attempting to upload the code to the board.
- Serial Monitor: Accessible via the toolbar, this allows you to view and send data between the Arduino and your computer. It is useful for debugging and monitoring real-time data.

4. Sketch Area

- Setup Function: This function runs once when the program starts. It's used for initializing variables, pin modes, etc.
- Loop Function: This function runs continuously after the setup function. It contains the main code that controls the Arduino's behavior.

5. Library Management

- Library Manager: Accessible through the Sketch menu, it allows users to include and manage libraries, which are collections of code that simplify complex tasks and expand the Arduino’s capabilities.

6. Board and Port Selection

- Board Selector: Located in the Tools menu, this lets you select the specific Arduino board you are using, ensuring that the code is compiled for the correct hardware.
- Port Selector: This allows you to select the correct port for uploading the code to your Arduino board, which is essential for communication between the computer and the board.

7. Examples and Documentation

- Examples: Accessible from the File menu, it provides a variety of pre-written sketches that demonstrate how to use specific functions and libraries. This is especially useful for beginners.
- Help Menu: Contains links to the official Arduino documentation, tutorials, and forums, providing additional resources for troubleshooting and learning.

3. IMPLEMENTATION

3.1 SOURCE CODE

```
const int A=20; // cm
const int B=15; // cm

const int trigPinX=2, echoPinX=3;
const int trigPinY=5, echoPinY=6;
const int servoPinL=8;
const int servoPinR=10;

float measureDist(int trigPin, int echoPin)
{
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(20);
  digitalWrite(trigPin, LOW);
  return pulseIn(echoPin, HIGH)/2*.0343;
}

void servoControl(int servoPin, int angle)
{
  unsigned int duration;
  duration=map(angle, 0, 180, 544, 2400);
  digitalWrite(servoPin, HIGH);
  delayMicroseconds(duration);
  digitalWrite(servoPin, LOW);
}
```

```
void setup()
{
  pinMode(trigPinX, OUTPUT);
  pinMode(trigPinY, OUTPUT);
  pinMode(servoPinL, OUTPUT);
  pinMode(servoPinR, OUTPUT);
}

void loop()
{
  float distX, distY;
  int pos, diff;
  int limDistX, limDistY;

  distX=measureDist(trigPinX, echoPinX);
  limDistX=(int)constrain(distX, 4, 2*A);
  pos=map(limDistX, 4, 2*A, 45, 135);

  distY=measureDist(trigPinY, echoPinY);
  limDistY=(int)constrain(distY, 4, B);
  diff=map(limDistY, 4, B, 40, 0);

  if(pos==135)
  {
    pos=90;
    diff=0;
  }
```

```
    servoControl(servoPinL, pos-diff);

    servoControl(servoPinR, pos+diff);

    delay(60);

  }
```

3.2 RESULTS

Figure 3.2.1 Sample Output-1: Object in Left

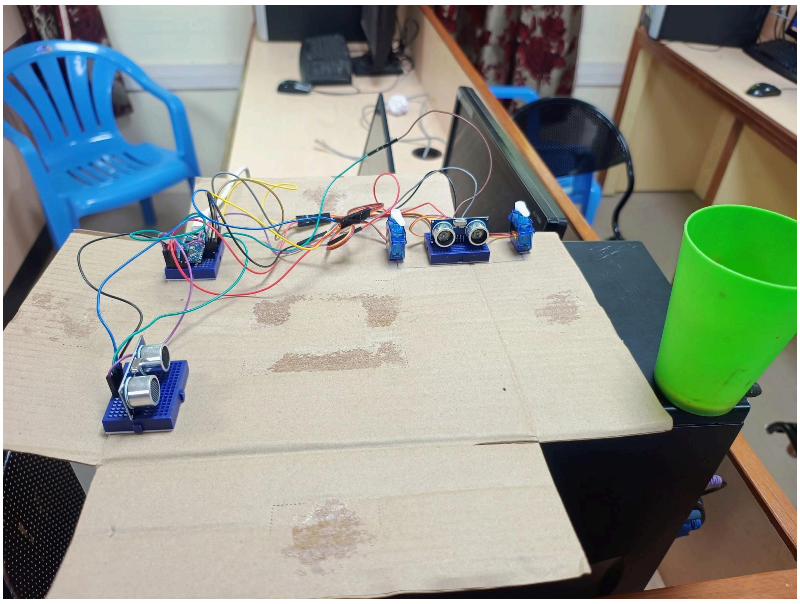

Figure 3.2.2 Sample Output-2: Object in Right

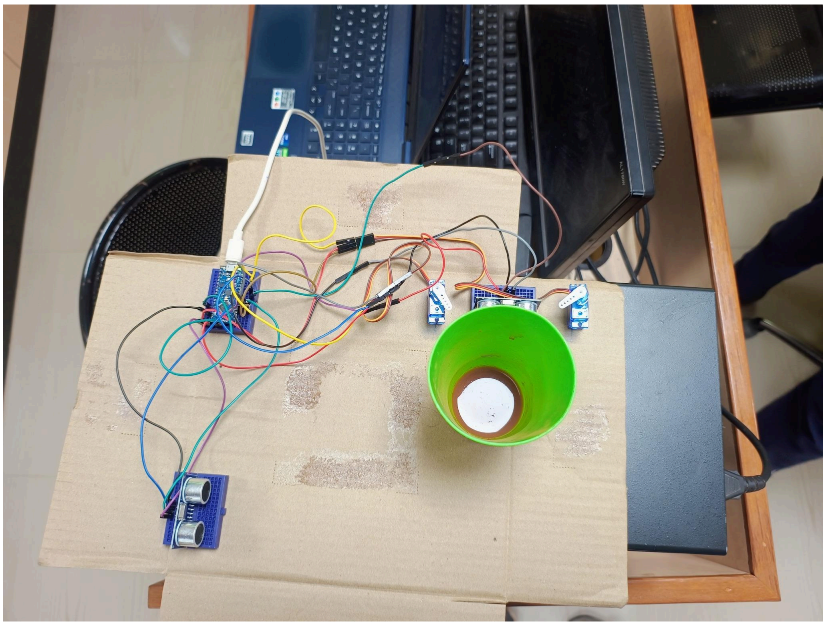

Figure 3.2.3 Sample Output-3: Object in Close

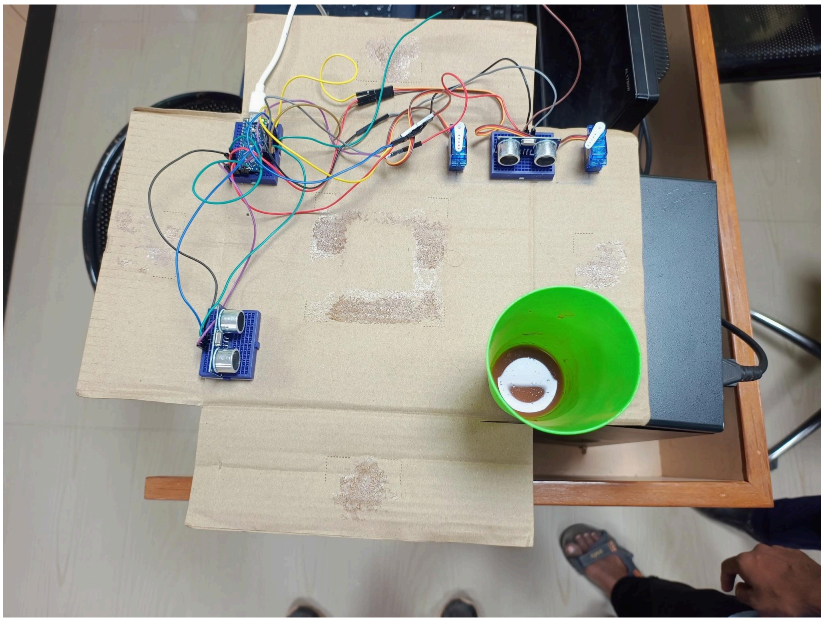

Figure 3.2.4 Sample Output-4: Object in Straight

4. CONCLUSION

4.1 CONCLUSION

The Object Tracking System provides a robust and efficient solution for real-time object monitoring using ultrasonic sensors and servo motors. The system measures the distance and position of objects and dynamically adjusts servo movements to track the target. The use of an Arduino Nano Every ensures that the system remains cost-effective and accessible for prototyping and implementation.

This project demonstrates vast potential in applications such as security, automation, and robotics, making it versatile and adaptable for different industries. With real-time tracking, it ensures effective monitoring and offers scalability for future modifications.

Future improvements, such as integrating more advanced tracking algorithms, expanding sensor range, or enabling wireless communication for remote control, could further enhance the system's capabilities. This could lead to broader applications in smart home systems, robotics, and autonomous vehicles, offering a practical solution for various object tracking scenarios.

5. REFERENCE

5.1 REFERENCE

- https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm
- https://learn.adafruit.com/ultrasonic-sensors
- https://docs.arduino.cc/learn/microcontrollers/analog-output
- https://docs.arduino.cc/hardware/nano-every/