Project Report

Student Name: Alaina Verma **UID: 23MCC20028**
Branch: MCA - CCD **Section/Group: 23MCD-2**
Semester: 4Th **Date of Performance: 11/04/2025**
Subject Name: Internet of Things **Subject Code: 23CAH-702**

Aim: IoT Based project to design and develop an Arduino-based Human-Following Robot

(Times New Roman-14)

Objectives: (Min. 3) (Times New Roman-14)

1. To design a human-following robot using an Arduino Uno as the central control unit. To integrate ultrasonic and infrared (IR) sensors for detecting the presence and movement of a human hand or body.
2. To develop a motor-controlled robotic platform using TT gear motors and a motor driver shield for movement in various directions. To implement real-time obstacle detection and distance measurement using the ultrasonic sensor.
3. To explore IoT-based sensor interfacing and automation principles in a mini robotics application. To enable directional-making based on sensor inputs to follow the human efficiently. To demonstrate autonomous navigation without manual control.

Components Required: (Times New Roman-14)

Sno	Name of Component	Qty.
1.	Arduino Uno (Times New Roman-12)	1
2.	Motor Driver Shield	1
3.	TT Gear Motors	4
4.	Wheels	4
5.	Ultrasonic Sensor	1
6.	Infrared Sensor	2

7.	Servo Motor	1
8.	Chassis (robot car body)	1
9.	Lithium-Ion Battery with holder	2
10.	Jumper Wires	10

Details of Components: (Times New Roman-14)

1. Arduino Uno: (Times New Roman-14)

Arduino Uno is an open-source microcontroller development board based on the ATmega328P microcontroller. It is one of the most popular boards used for electronic project, IoT, robotic and automation system due to its simplicity, versatility and ease of use. It helps to upload code easily and test new ideas quickly. It is cost-effective compared to other boards and free IDE, libraries and tons. It can handle multiple sensors and actuators simultaneously.

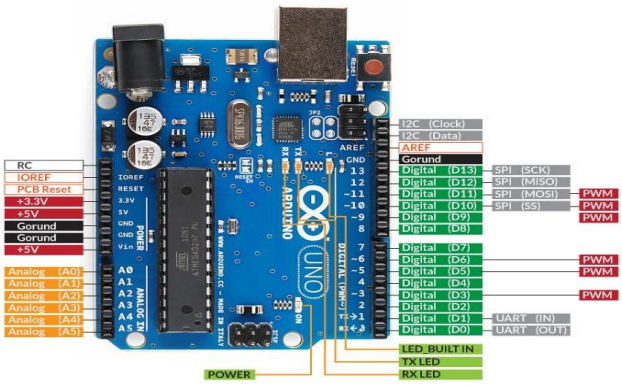
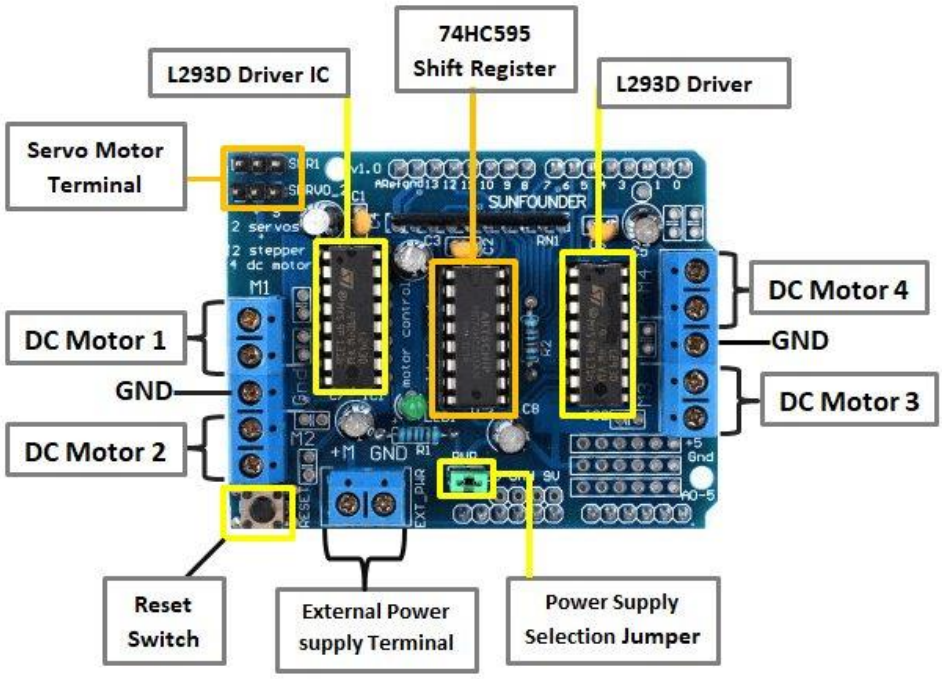


Figure 1 : NodeMCU Board

2. Motor Driver Shield: (Times New Roman-14)

A Motor Driver Shield is an electronic circuit board that acts as an interface between a microcontroller (Arduino) and Dc motors, servo motors or stepper motors. It allows the microcontroller to control direction and speed of motors by providing enough current an voltage which Arduino alone cannot do.

The most common motor driver IC used in Arduino projects is the L298N or L293D and in shield it sits directly on top of the Arduino board.



3. TT Gear: (Times New Roman-14)

A TT gear motor is a small DC motor with a built-in gearbox (gear reduction system), commonly used in small DIY and robotics projects. The gearbox helps reduce the motor's speed and increase its torque (rotational power), making it suitable for applications that require strength over speed.

4. Ultrasonic Sensor: (Times New Roman-14)

An ultrasonic sensor is an electronic device that measures the distance to an object by using ultrasonic sound waves. It works like a mini sonar system — similar to how bats or submarines detect objects.

The most commonly used sensor in Arduino projects is the HC-SR04 Ultrasonic Sensor.

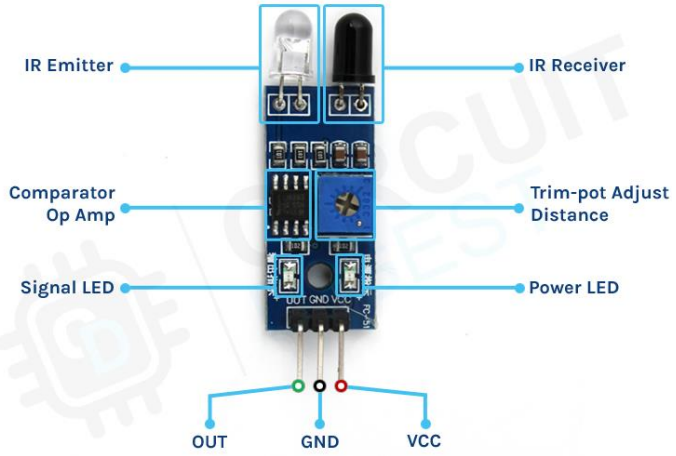
It sends out an ultrasonic wave (typically 40kHz) and waits for the echo to bounce back from an object. By calculating the time taken for the echo to return, the sensor can determine the distance of the object.

5. Infrared Sensor: (Times New Roman-14)

An Infrared (IR) sensor is an electronic device that detects infrared radiation (heat) emitted from objects. In Arduino projects, it's typically used to detect the presence of objects or motion by sensing reflected IR light.

There are two main types:

- **Active IR Sensor:** Emits and detects IR light (commonly used in Arduino).
- **Passive IR Sensor (PIR):** Detects heat radiation from bodies (used in motion detection, not in your project).



6. Servo Motor: (Times New Roman-14)

A ervo motor is a type of motor that allows precise control of angular position. It is widely used in robotics and automation systems where specific rotation angles are needed.
Unlike regular motors, a servo motor rotates to a specific angle between 0° and 180° (or sometimes 360° for continuous ones), based on the control signal received.

Block Diagram of Designed Model:

Explanation of Block Diagram-(Times New Roman -12)

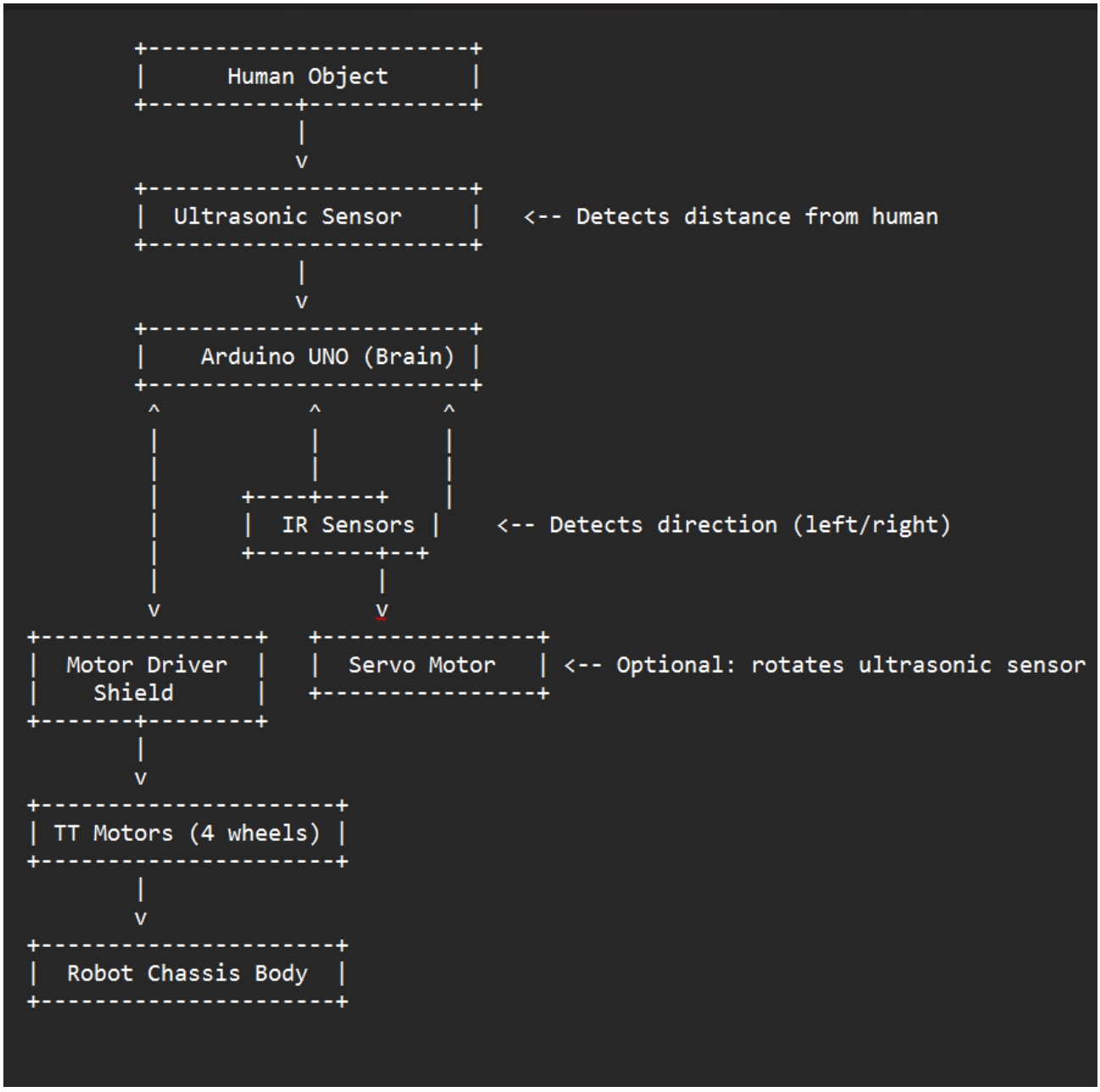


Figure 2 : Block Diagram of Model

Working of Designed Model:

- As mentioned in Figure 2. Smart home devices, including lights, thermostats, cameras, locks, and appliances, are equipped with sensors and connected to the home's WiFi network, allowing them to communicate with each other and be controlled remotely.
- Some systems use alternative protocols like Zigbee, Z-Wave, or Bluetooth for connectivity, depending on the device and its requirements. Smart home devices, including lights, thermostats, cameras, locks, and appliances, are equipped with sensors and connected to the home's WiFi network, allowing them to communicate with each other and be controlled remotely.
- Some systems use alternative protocols like Zigbee, Z-Wave, or Bluetooth for connectivity, depending on the device and its requirements. Smart home devices, including lights, thermostats, cameras, locks, and appliances, are equipped with sensors and connected to the home's WiFi network, allowing them to communicate with each other and be controlled remotely. Some systems use alternative protocols like Zigbee, Z-Wave, or Bluetooth for connectivity, depending on the device and its requirements.

Pictures of Prototype:

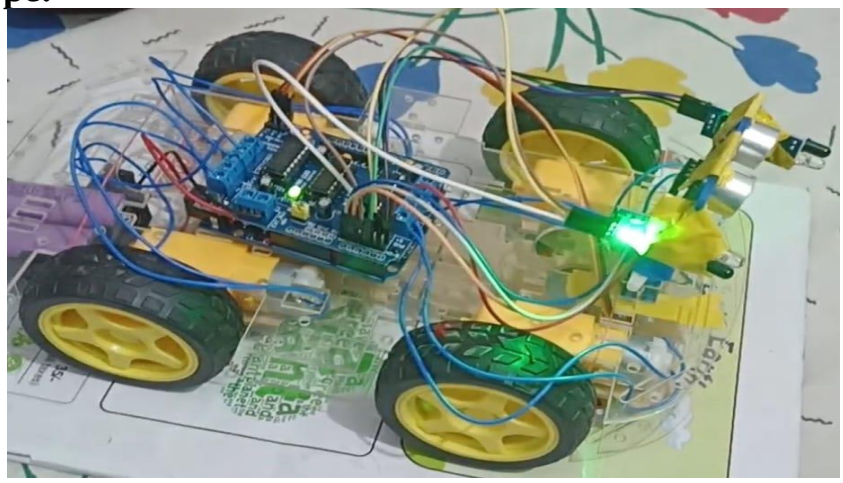


Figure 3 : Designed Model

```
/*
 * Hand-Following Robot
 *
 * This robot uses ultrasonic and IR sensors to detect and follow a hand.
 * When it detects a hand at an appropriate distance, it automatically follows it.
 *
 * Hardware:
 * - Arduino Uno/Nano
 * - L293D Motor Driver Shield
 * - Ultrasonic Sensor (HC-SR04)
 * - 2x IR Sensors
 * - 4x TT Gear Motors
 * - SG90 Servo Motor (for scanning)
 *
 * Connections:
 * - Ultrasonic: Trigger->A1, Echo->A0
 * - IR Sensors: Right->A2, Left->A3
 * - Motors connected through Motor Shield
 * - Servo: Signal->D10
 */

#include <NewPing.h>      // For ultrasonic sensor
#include <Servo.h>        // For servo motor
#include <AFMotor.h>      // For Adafruit/generic L293D motor shield

// Pin Definitions
#define TRIGGER_PIN  A1   // Ultrasonic sensor trigger pin
#define ECHO_PIN     A0   // Ultrasonic sensor echo pin
#define RIGHT_IR     A2   // Right IR sensor pin
#define LEFT_IR      A3   // Left IR sensor pin
#define SERVO_PIN    10   // Servo motor pin

// Constants
#define MAX_DISTANCE     100  // Maximum ultrasonic sensing distance (cm)
#define MIN_DISTANCE     10   // Minimum follow distance (too close, stop)
#define OPTIMAL_DISTANCE 20   // Target distance to maintain from hand
#define MAX_FOLLOW       50   // Maximum distance to start following
#define MOTOR_SPEED      255  // Base motor speed (0-255)
```

```
#define TURN_SPEED       255  // Speed when turning

// Setup objects for sensors and motors
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
Servo scanServo;

// Create motor objects - using all 4 channels
AF_DCMotor frontRight(1, MOTOR12_1KHZ); // Front right motor on M1
AF_DCMotor frontLeft(2, MOTOR12_1KHZ);  // Front left motor on M2
AF_DCMotor backRight(3, MOTOR34_1KHZ);  // Back right motor on M3
AF_DCMotor backLeft(4, MOTOR34_1KHZ);   // Back left motor on M4

// Variables
unsigned int distance = 0;    // Distance from ultrasonic
bool rightDetected = false;   // IR sensor detection flags
bool leftDetected = false;
unsigned long lastScanTime = 0;  // For periodic scanning when idle

void setup() {
  // Initialize serial communication
  Serial.begin(9600);
  Serial.println("Hand-Following Robot Initializing...");

  // Initialize IR sensors as inputs
  pinMode(RIGHT_IR, INPUT);
  pinMode(LEFT_IR, INPUT);

  // Initialize servo
  scanServo.attach(SERVO_PIN);
  scanServo.write(90); // Center position

  // Initialize motors (start with stopped)
  frontRight.setSpeed(0);
  frontLeft.setSpeed(0);
  backRight.setSpeed(0);
  backLeft.setSpeed(0);

  stopMotors();

  // Initial scan to show the robot is ready
```

```
  scanForHand();

  Serial.println("Robot Ready!");
  delay(1000);
}

void loop() {
  // Read sensors
  readSensors();

  // Main decision logic for following hand
  if (distance > 0) { // If distance is 0, sensor detected nothing
    if (distance < MIN_DISTANCE) {
      // Hand too close - stop
      Serial.println("Hand too close - stopping");
      stopMotors();
    }
    else if (distance <= MAX_FOLLOW) {
      // Hand in follow range - determine direction
      if (!rightDetected && leftDetected) {
        // Hand detected on left side
        Serial.println("Hand on left - turning left");
        turnLeft();
      }
      else if (rightDetected && !leftDetected) {
        // Hand detected on right side
        Serial.println("Hand on right - turning right");
        turnRight();
      }
      else {
        // Hand centered or both IR sensors triggered
        // Adjust speed based on distance
        if (distance < OPTIMAL_DISTANCE) {
          // Slightly closer than optimal - slow down
          Serial.println("Hand centered but close - moving slowly");
          moveForward(MOTOR_SPEED - 50);
        }
        else if (distance > OPTIMAL_DISTANCE + 10) {
          // Further than optimal - speed up
          Serial.println("Hand centered but far - moving faster");
```

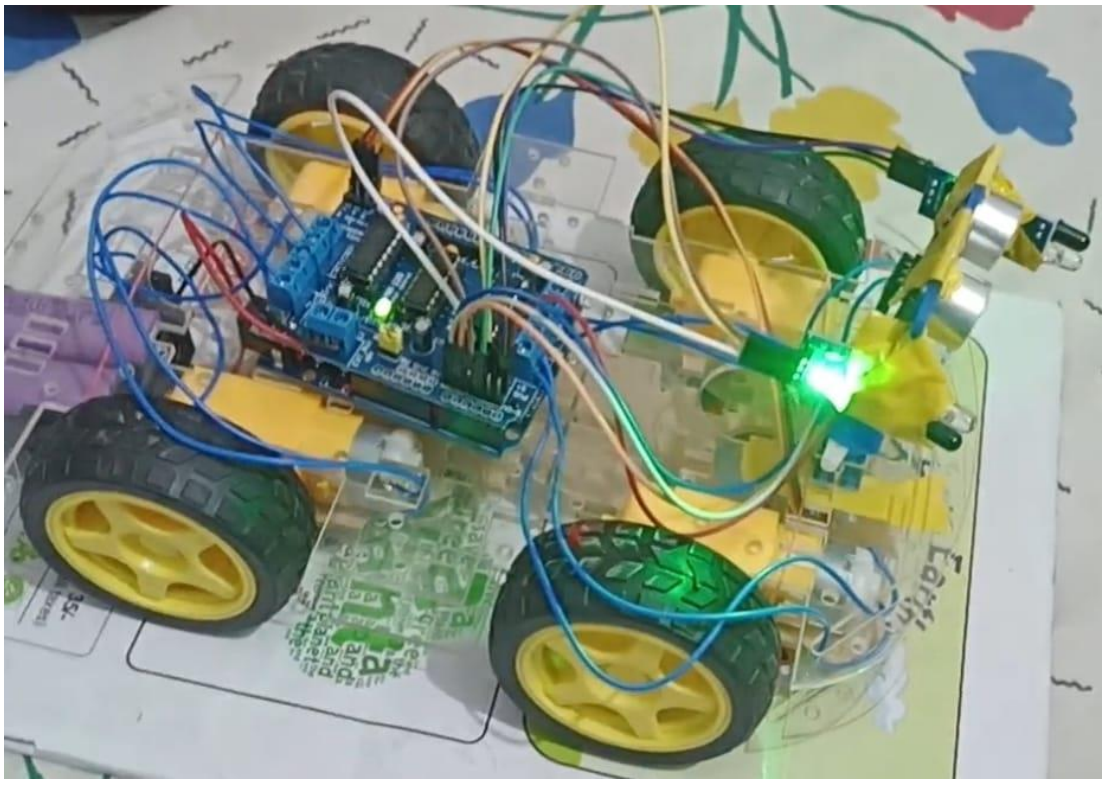
```
      moveForward(MOTOR_SPEED + 30);
     }
     else {
      // At optimal distance - maintain speed
      Serial.println("Hand centered at optimal distance - following");
      moveForward(MOTOR_SPEED);
     }
    }
   }
   else {
    // Hand too far away - stop and scan
    Serial.println("Hand too far - stopping");
    stopMotors();

    // Periodically scan when idle
    if (millis() - lastScanTime > 3000) { // Every 3 seconds
     scanForHand();
     lastScanTime = millis();
    }
   }
  }
  else {
   // No hand detected - stop and scan
   Serial.println("No hand detected - stopping");
   stopMotors();

   // Periodically scan when idle
   if (millis() - lastScanTime > 3000) { // Every 3 seconds
    scanForHand();
    lastScanTime = millis();
   }
  }

  // Small delay for stability
  delay(50);
}

void readSensors() {
  // Read ultrasonic sensor (in cm)
  distance = sonar.ping_cm();
```

```
  // Read IR sensors (typically LOW when object detected, HIGH when nothing detected)
  // But this might be reversed for some sensors - adjust as needed
  rightDetected = (digitalRead(RIGHT_IR) == LOW);
  leftDetected = (digitalRead(LEFT_IR) == LOW);

  // Debug output
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.print("cm | Right IR: ");
  Serial.print(rightDetected ? "Detected" : "None");
  Serial.print(" | Left IR: ");
  Serial.println(leftDetected ? "Detected" : "None");
}

void moveForward(int speed) {
  frontRight.setSpeed(speed);
  frontLeft.setSpeed(speed);
  backRight.setSpeed(speed);
  backLeft.setSpeed(speed);

  frontRight.run(FORWARD);
  frontLeft.run(FORWARD);
  backRight.run(FORWARD);
  backLeft.run(FORWARD);
}

void turnRight() {
  frontRight.setSpeed(TURN_SPEED);
  frontLeft.setSpeed(TURN_SPEED);
  backRight.setSpeed(TURN_SPEED);
  backLeft.setSpeed(TURN_SPEED);

  frontRight.run(BACKWARD);
  frontLeft.run(FORWARD);
  backRight.run(BACKWARD);
  backLeft.run(FORWARD);
}

void turnLeft() {
```

```
  frontRight.setSpeed(TURN_SPEED);
  frontLeft.setSpeed(TURN_SPEED);
  backRight.setSpeed(TURN_SPEED);
  backLeft.setSpeed(TURN_SPEED);

  frontRight.run(FORWARD);
  frontLeft.run(BACKWARD);
  backRight.run(FORWARD);
  backLeft.run(BACKWARD);
}

void stopMotors() {
  frontRight.run(RELEASE);
  frontLeft.run(RELEASE);
  backRight.run(RELEASE);
  backLeft.run(RELEASE);
}

void scanForHand() {
  Serial.println("Scanning for hand...");

  // Scan from center to right
  for (int pos = 90; pos <= 180; pos += 5) {
    scanServo.write(pos);
    delay(30);

    // Check for hand during scan
    if (sonar.ping_cm() > 0 && sonar.ping_cm() < MAX_FOLLOW) {
      Serial.println("Hand detected during scan!");
      scanServo.write(90); // Return to center
      return;
    }
  }

  // Scan from right to left
  for (int pos = 180; pos >= 0; pos -= 5) {
    scanServo.write(pos);
    delay(30);

    // Check for hand during scan
```

```
  if (sonar.ping_cm() > 0 && sonar.ping_cm() < MAX_FOLLOW) {
    Serial.println("Hand detected during scan!");
    scanServo.write(90); // Return to center
    return;
  }
 }

 // Return to center
 scanServo.write(90);
 lastScanTime = millis(); // Update scan time
}
```

Output of Deigned Model/Prototype (Paste Images of output)



Learning outcomes (What I have learnt):

1. **Understanding Sensor Integration and Control Systems:**
   Gained hands-on experience in interfacing ultrasonic and infrared (IR) sensors with the Arduino Uno to detect human presence and guide robotic movement.
2. **Motor Control and Hardware Implementation:**
   Acquired practical skills in assembling and programming TT gear motors with a motor driver shield for directional movement and obstacle avoidance.
3. **IoT Concepts in Mini Robotics:**
   Explored key Internet of Things (IoT) concepts through real-time data collection, decision-making, and automation in a robotics application.
4. **Problem-Solving and Debugging Skills:**
   Developed analytical and troubleshooting abilities while working with hardware connections, sensor calibration, and logic implementation in Arduino IDE.