A red and blue text on a white background

Description automatically generated

**HUMAN FOLLOWER**

**ROBOT**

**AI-B**

D. Sravan(21911A3578)

D. Srujana(21911A3579)

G.Rajeshwari(21911A3580)

G. Sri Charan(21911A3581)

K.Spandana(21911A3593)

**Overview**

A human follower robot autonomously tracks and follows a human user using sensors like LIDAR, cameras, and infrared. It navigates complex environments, avoiding obstacles and maintaining a set distance. Applications include personal assistance, healthcare, logistics, and security. Challenges involve improving robustness, safety, and battery life. Future advancements will enhance AI, sensor tech, and human-robot interaction.

**Components**

1. Arduino Uno
2. Motor Driver Shield
3. BO Motor and Wheel
4. Ultrasonic Sensor
5. Servo Motor
6. IR Sensor
7. Jumper Wire
8. 18650 Battery
9. Battery Pocket
10. Sun Board

**Assembly Instructions**

**Prepare the Components:**

* Gather the necessary components, including an Arduino board (such as Arduino Uno), IR sensors, ultrasonic sensor, servo motor, motor driver module (e.g., L298N), wheels, chassis, and a power source (usually a battery).
* You’ll also need jumper wires, a breadboard (if required), and double-sided tape or hot glue for mounting.

**Build the Chassis:**

* Assemble the robot’s chassis according to your design. You can use acrylic sheets, 3D-printed parts, or any other material.
* Attach the wheels to the motors and secure them to the chassis.

**Mount the Components:**

* Place the Arduino board in the centre of the chassis. Attach the ultrasonic sensor (for distance measurement) and IR sensors (for hand detection) to the front of the robot.
* Mount the servo motor (used for steering) on the chassis.

**Wiring:**

* Connect the components :
* Ultrasonic sensor: Connect VCC, GND, and signal pins to the Arduino.
* IR sensors: Connect them to the Arduino’s digital pins.
* Servo motor: Connect it to a PWM (pulse-width modulation) pin on the Arduino.
* Motor driver module: Connect it to the Arduino for controlling the motors.

**Fine-Tuning:**

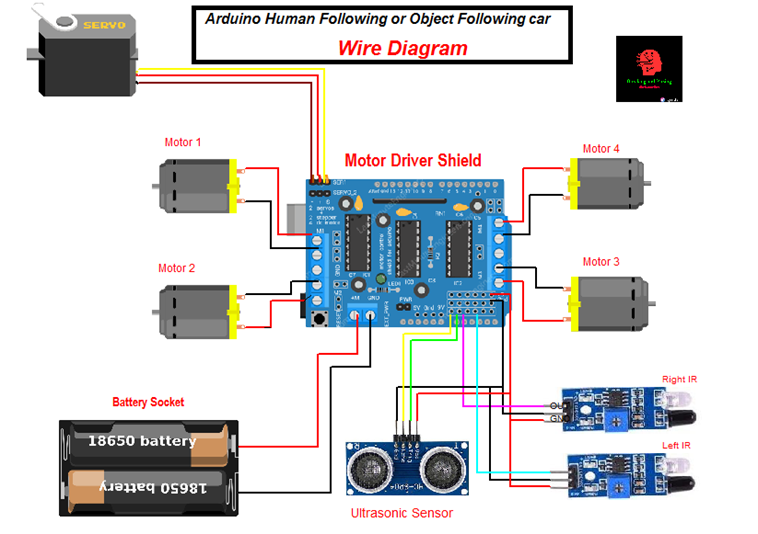
* Calibrate the ultrasonic sensor to measure distances accurately.
* Adjust the servo motor’s angle to steer the robot smoothly.
* Test the robot in different environments to ensure reliable performance.

**Testing:**

* Power up the robot using the battery. Place your hand in front of the IR sensors to test if the robot follows it.
* Adjust the code and sensor positions as needed to improve accuracy.

**Circuit Diagram**

Refer to the following image for the complete circuit diagram. It includes connections between Battery, Arduino Board, Servo Motor, IR Sensor, Driver shield and Motor Wheels.



**Arduino Code**

#include<NewPing.h>

#include<Servo.h>

#include<AFMotor.h>

#define RIGHT A2

#define LEFT A3

#define TRIGGER\_PIN A1

#define ECHO\_PIN A0

#define MAX\_DISTANCE 100

NewPing sonar(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANCE);

AF\_DCMotor Motor1(1,MOTOR12\_1KHZ);

AF\_DCMotor Motor2(2,MOTOR12\_1KHZ);

AF\_DCMotor Motor3(3,MOTOR34\_1KHZ);

AF\_DCMotor Motor4(4,MOTOR34\_1KHZ);

Servo myservo;

int pos =0;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

myservo.attach(10);

{

for(pos = 90; pos <= 180; pos += 1){

myservo.write(pos);

delay(15);

}

for(pos = 180; pos >= 0; pos-= 1) {

myservo.write(pos);

delay(15);

}

for(pos = 0; pos<=90; pos += 1) {

myservo.write(pos);

delay(15);

}

}

pinMode(RIGHT, INPUT);

pinMode(LEFT, INPUT);

}

void loop() {

// put your main code here, to run repeatedly:

delay(50);

unsigned int distance = sonar.ping\_cm();

Serial.print("distance");

Serial.println(distance);

int Right\_Value = digitalRead(RIGHT);

int Left\_Value = digitalRead(LEFT);

Serial.print("RIGHT");

Serial.println(Right\_Value);

Serial.print("LEFT");

Serial.println(Left\_Value);

if((Right\_Value==1) && (distance>=10 && distance<=30)&&(Left\_Value==1)){

Motor1.setSpeed(160);

Motor1.run(FORWARD);

Motor2.setSpeed(160);

Motor2.run(FORWARD);

Motor3.setSpeed(160);

Motor3.run(FORWARD);

Motor4.setSpeed(160);

Motor4.run(FORWARD);

}

else if((Right\_Value==0) && (Left\_Value==1)) {

Motor1.setSpeed(250);

Motor1.run(FORWARD);

Motor2.setSpeed(250);

Motor2.run(FORWARD);

Motor3.setSpeed(150);

Motor3.run(BACKWARD);

Motor4.setSpeed(150);

Motor4.run(BACKWARD);

}

else if((Right\_Value==1)&&(Left\_Value==0))

{

Motor1.setSpeed(150);

Motor1.run(BACKWARD);

Motor2.setSpeed(150);

Motor2.run(BACKWARD);

Motor3.setSpeed(250);

Motor3.run(FORWARD);

Motor4.setSpeed(250);

Motor4.run(FORWARD);

}

else if((Right\_Value==1)&&(Left\_Value==1))

{

Motor1.setSpeed(0);

Motor1.run(RELEASE);

Motor2.setSpeed(0);

Motor2.run(RELEASE);

Motor3.setSpeed(0);

Motor3.run(RELEASE);

Motor4.setSpeed(0);

Motor4.run(RELEASE);

}

else if(distance > 1 && distance < 10)

{

Motor1.setSpeed(0);

Motor1.run(RELEASE);

Motor2.setSpeed(0);

Motor2.run(RELEASE);

Motor3.setSpeed(0);

Motor3.run(RELEASE);

Motor4.setSpeed(0);

Motor4.run(RELEASE);

}

}

**Usage Instructions**

1. **Setup:** Unbox the robot, charge it fully, power on, and calibrate the sensors for the correct following distance.
2. **Pairing:** Attach the wearable tag to the person, manually pair the robot via the control panel or mobile app.
3. **Operation:** Activate follow mode, start walking, and the robot will maintain the set distance and avoid obstacles.

**Maintenance**

* Regularly clean the sensors and wheels for optimal performance.
* Update the firmware through the mobile app to ensure the latest features and fixes.
* Monitor battery health, avoid overcharging, and perform a full discharge monthly to maintain battery life.

**Trouble Shooting**

* **Robot Not Following:** Ensure the wearable tag is properly positioned and the robot is in follow mode; recalibrate if necessary.
* **Loss of Connection:** Check Bluetooth/Wi-Fi connections and ensure both the robot and app are up to date.
* **Obstacle Detection Issues:** Clean the sensors and check for firmware updates to enhance sensor performance.

Top of Form

Bottom of Form