SainSmart 4WD Mobile Car-Ultrasonic Obstacle Avoidance

www.sainsmart.com

Manual:

1. Weld leads on the four ports of the DC geared motor.

2. Assemble the 4WD Mobile Robot Platform.

3. Fix the Dc Motor Driver Board on the holes of the inside body of the car.

4. Wiring:

1) The right-front wheel's lead: The negative pole is above, the positive pole is below.
The negative pole connects to OUT1 port
The positive pole connects OUT2 port;

2) The rear-right wheel's lead: The negative pole is above, the positive pole is below.
The **negative pole** connects to **OUT2 port**
The **positive pole** connects to **OUT1 port**.

3) The left-front wheel's lead: The positive pole is above, the negative pole is below.
The **negative** pole connects to **OUT4 port**
The **positive** pole connects to **OUT3 port**.

4) The left-rear wheel's lead: The positive pole is above, the negative pole is below.
The **negative pole** connects to **OUT3 port**
The **positive pole** connects to **OUT4 port.**

5. Connect Sensor V5 to UNO. Connect **IN1, IN2, IN3, IN4** to the **8,9,6,7 pin** of the Sensor V5 accordingly.

6. Connect Power Supply *Interface* **5V and GND** port of the motor driver board to any **V and G** pin of the Sensor V5 board.

7. Connect power supply interface VCC and GND on the motor driver board to 9-35V battery or Dc power supply.

8. Fix the HC-SR04 and UNO onto the car.

9. Connect **VCC** and **GND** port of the HC-SR04 to any **V and G pin** of the sensor V5. **Trig** and **Echo** port connects to **3, 2 port** of the Sensor V5 board accordingly.

10. Download the testing program to the UNO to get the car moving and automatically avoid obstacles.

Testing Program:

```
const int EchoPin = 2; // Ultrasonic signal input

const int TrigPin = 3; // Ultrasonic signal output


const int leftmotorpin1 = 8; //signal output of Dc motor driven plate

const int leftmotorpin2 = 9;

const int rightmotorpin1 = 6;

const int rightmotorpin2 = 7;


const int HeadServopin = 10; // signal input of headservo

const int Sharppin = 11; // infrared output


const int maxStart = 800;     //run dec time


// Variables

int isStart = maxStart;        //start

  int currDist = 0;        // distance

  boolean running = false;


void setup() {


    Serial.begin(9600); // Serial begin texting
```

```
    //signal input port
    pinMode(EchoPin, INPUT);
    pinMode(Sharppin, INPUT);

    //signal output port
    for (int pinindex = 3; pinindex < 11; pinindex++) {
        pinMode(pinindex, OUTPUT); // set pins 3 to 10 as outputs
    }

    // headservo interface
    headservo.attach(HeadServopin);

    //start buffer movable head
    headservo.write(70);
    delay(2000);
    headservo.write(20);
    delay(2000);
}

void loop() {
```

```
if(DEBUG){
    Serial.print("running:");
    if(running){
        Serial.println("true");
    }
    else{
        Serial.println("false");
    }
}

if (isStart <= 0) {
    if(running){
        totalhalt();        // stop!
    }
    int findsomebody = digitalRead(Sharppin);
    if(DEBUG){
        Serial.print("findsomebody:");
        Serial.println(findsomebody);
    }
    if(findsomebody > 0) {
        isStart = maxStart;
    }
```

```
    delay(4000);
    return;
}
isStart--;
delay(100);

if(DEBUG){
    Serial.print("isStart: ");
    Serial.println(isStart);
}

currDist = MeasuringDistance(); //measure front distance

if(DEBUG){
    Serial.print("Current Distance: ");
    Serial.println(currDist);
}

if(currDist > 30) {
    nodanger();
}
else if(currDist < 15){
```

```
        backup();
        randTrun();
    }
    else {
        //whichway();
        randTrun();
    }
}

//measure distance, unit “cm”
long MeasuringDistance() {
    long duration;
    //pinMode(TrigPin, OUTPUT);
    digitalWrite(TrigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(TrigPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(TrigPin, LOW);

    //pinMode(EchoPin, INPUT);
    duration = pulseIn(EchoPin, HIGH);
```

```
    return duration / 29 / 2;
}

// forward
void nodanger() {
    running = true;
    digitalWrite(leftmotorpin1, HIGH);
    digitalWrite(leftmotorpin2, LOW);
    digitalWrite(rightmotorpin1, HIGH);
    digitalWrite(rightmotorpin2, LOW);
    return;
}

//backward
void backup() {
    running = true;
    digitalWrite(leftmotorpin1, LOW);
    digitalWrite(leftmotorpin2, HIGH);
    digitalWrite(rightmotorpin1, LOW);
    digitalWrite(rightmotorpin2, HIGH);
    delay(1000);
}
```

```
//choose way
  void whichway() {
    running = true;
    totalhalt();        // first stop!

    headservo.write(20);
    delay(1000);
    int lDist = MeasuringDistance();      // check left distance
    totalhalt();            // probe recovering

    headservo.write(120);    // turn the servo right
    delay(1000);
    int rDist = MeasuringDistance();      // check right distance
    totalhalt();            // probe recovering

    if(lDist < rDist) {
        body_lturn();
    }
    else{
        body_rturn();
    }
```

```
    return;
}

//remodulate to the original status mechanically
void totalhalt() {
    digitalWrite(leftmotorpin1, HIGH);
    digitalWrite(leftmotorpin2, HIGH);
    digitalWrite(rightmotorpin1, HIGH);
    digitalWrite(rightmotorpin2, HIGH);
    headservo.write(70);    //    set servo to face forward
    running = false;
    return;
    delay(1000);
}

//turn left
void body_lturn() {
    running = true;
    digitalWrite(leftmotorpin1, LOW);
    digitalWrite(leftmotorpin2, HIGH);
    digitalWrite(rightmotorpin1, HIGH);
    digitalWrite(rightmotorpin2, LOW);
```

```
    delay(1500);
    totalhalt();
}

//turn right
void body_rturn() {
    running = true;
    digitalWrite(leftmotorpin1, HIGH);
    digitalWrite(leftmotorpin2, LOW);
    digitalWrite(rightmotorpin1, LOW);
    digitalWrite(rightmotorpin2, HIGH);
    delay(1500);
    totalhalt();
}

void randTrun(){
    long randNumber;
    randomSeed(analogRead(0));
    randNumber = random(0, 10);
    if(randNumber > 5) {
        body_rturn();
    }
```

```
    else
    {
        body_lturn();
    }
}
```

Items:

SainSmart UNO: SainSmart UNO ATMEGA328P-PU ATMEGA8U2 Microcontroller For Arduino SainSmart

SainSmart Sensor Shield V5:

http://www.sainsmart.com/sainsmart-sensor-shield-v5-4-arduino-apc220-bluetooth-analog-module-servo-motor.html

SainSmart 4WD Mobile Robot Platform:

http://www.sainsmart.com/sainsmart-4wd-drive-aluminum-mobile-robot-platform-for-robot-arduino-uno-mega2560-r3-duemilanove-black.html

SainSmart L298N Dual H Bridge Stepper Motor Driver:

http://www.sainsmart.com/sainsmart-l298n-dual-h-bridge-stepper-motor-driver-controller-board-module-for-arduino-robot.html

Ultrasonic HC-SR04 Distance Sensor:

http://www.sainsmart.com/ultrasonic-ranging-detector-mod-hc-sr04-distance-sensor.html