

# robopro.ino

```
//libraries used
#include
#include
#include
const int rs = A0, en = A5, d4 = 13, d5 = 12, d6 = 8, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
//motor driver pins
const int LMForward = 7;
const int LMBackward = 6;
const int RMForward = 4;
const int RMBackward = 5;
//for counting shape
int cuboid=0;
int sphere=0;
boolean GoFo = false;//condition variable
int distance = 100;
//ultrasonic pins
#define Tpin1 A1
#define Epin1 A2
#define Tpin2 A3
#define Epin2 A4
#define maxdist 200//max distance for ultrasonic sensor

NewPing sonar[2]={NewPing(Tpin1, Epin1, maxdist),NewPing(Tpin2, Epin2, maxdist)};//newping sonar function
Servo Smotor; //servo motor
//enable pin for speed control
int en1=3;
int en2=11;

void setup(){
  pinMode(en1, OUTPUT);
  pinMode(en2, OUTPUT);
  pinMode(RMForward, OUTPUT);
  pinMode(LMForward, OUTPUT);
```

```

pinMode(LMBackward, OUTPUT);
pinMode(RMForward, OUTPUT);
pinMode(9, OUTPUT);
lcd.begin(16, 2);
lcd.clear();
Smotor.attach(10);
//setting servomotor to 90 degrees
Smotor.write(90);
delay(2000);
//stabalizing distance by taking twice
distance = RePing();
delay(100);
distance = RePing();
delay(100);
Serial.begin(9600);
//setting speed of bot
analogWrite(en1,60);
analogWrite(en2,60);

}

void loop(){
//setting contrast to lcd display
analogWrite(9,20);
//distance on right and left
int distRight = 0;
int distLeft = 0;
delay(50);
//if distance is less than 20 then shape detection is done by stoping
if (distance <= 20){
//stoping bot
Stopbot();
delay(200);
//counting shape
countshape();
delay(100);
moveBackward();

```

```

delay(400);
Stopbot();
delay(300);
distRight = senseRight();
delay(300);
distLeft = senseLeft();
delay(300);
//turning right if distace on right is more and vice versa
if (distRight > distLeft){
turnRight();
Stopbot();
}
else if(distLeft>distRight){
turnLeft();
Stopbot();
}
else{
moveBackward();
Stopbot();
}
}
else{
moveForward();
}
distance = RePing();//again getting distance after counting shape and detection
}
void countshape()
{
//getting values of both the sensors and comparing them
int cm1= sonar[0].ping_cm();
if (cm1==0)
cm1=250;

delay(200);
int cm2= sonar[1].ping_cm();
if (cm2==0)
cm2=250;

```

```
cm2+=2;
delay(200);
if(cm2==cm1||cm2==(cm1+1)||cm2==(cm1-1)||cm2==(cm1-2)||cm2==(cm1+2))//if values are in +-2 range then a cuboid else a
sphere
cuboid++;
else
sphere++;
//displaying values on lcd display
lcd.setCursor(0, 0);
lcd.print("sphere=");
lcd.print(sphere);
lcd.setCursor(0, 1);
lcd.print("cuboid=");
lcd.print(cuboid);
}
//different functions for different purpose
int senseRight(){
Smotor.write(0);
delay(500);
int distance = RePing();
delay(100);
Smotor.write(90);
return distance;
}

int senseLeft(){
Smotor.write(180);
delay(500);
int distance = RePing();
delay(100);
Smotor.write(90);
return distance;
delay(100);
}

int RePing(){
delay(70);
```

```
int cm = sonar[1].ping_cm();
if (cm==0){
cm=250;
}
return cm;
}
```

```
void Stopbot(){
```

```
digitalWrite(RMForward, LOW);
digitalWrite(LMForward, LOW);
digitalWrite(RMBackward, LOW);
digitalWrite(LMBackward, LOW);
}
```

```
void moveForward(){
```

```
if(!GoFo){
```

GoFo=true;//GoFo function is used so that bot does stop frequently on forward function call everytime

```
digitalWrite(LMForward, HIGH);
digitalWrite(RMForward, HIGH);
```

```
digitalWrite(LMBackward, LOW);
digitalWrite(RMBackward, LOW);
}
}
```

```
void moveBackward(){
```

```
GoFo=false;
```

```
digitalWrite(LMBackward, HIGH);
digitalWrite(RMBackward, HIGH);
```

```
digitalWrite(LMForward, LOW);
```

```
digitalWrite(RMForward, LOW);

}

void turnRight(){

digitalWrite(LMForward, HIGH);
digitalWrite(RMBackward, HIGH);

digitalWrite(LMBackward, LOW);
digitalWrite(RMForward, LOW);

delay(500);

digitalWrite(LMForward, HIGH);
digitalWrite(RMForward, HIGH);

digitalWrite(LMBackward, LOW);
digitalWrite(RMBackward, LOW);


}

void turnLeft()
{

digitalWrite(LMBackward, HIGH);
digitalWrite(RMForward, HIGH);

digitalWrite(LMForward, LOW);
digitalWrite(RMBackward, LOW);

delay(500);

digitalWrite(LMForward, HIGH);
digitalWrite(RMForward, HIGH);
```

```
digitalWrite(LMBackward, LOW);  
digitalWrite(RMBackward, LOW);  
}
```