```
/*
L=left
R=right
F=forward
B=backward

*/
#include <Servo.h>

int pinLB=6; // define 6 pin left backward int pinLF=9; // define 9
pin left forward int pinRB=3; // define 10 pin right backward int
pinRF=5; // define 11 pin right forward int inputPin = A0; //
define ultrasonic signal receiving pin int outputPin = A1; // define
ultrasonic signal transmission pin
int Fspeedd = 0; // forward speed int
Rspeedd = 0; // turning right speed int
Lspeedd = 0; // turning left speed
int directionn = 0; // forward=8 backward=2 left=4 right=6 Servo
myservo; // set myservo
int delay_time = 250; // stabilizing time of servo motor after turning int
Fgo = 8; // forward int Rgo = 6; // turn right int Lgo = 4; // turn left int
Bgo = 2; // backward



void setup()
{
Serial.begin(9600); // define motor output pin
pinMode(pinLB,OUTPUT); // pin 8 (PWM)
pinMode(pinLF,OUTPUT); // pin 9 (PWM)
pinMode(pinRB,OUTPUT); // pin 10 (PWM)
pinMode(pinRF,OUTPUT); // pin 11 (PWM) pinMode(inputPin,
INPUT); // define ultrasonic input pin pinMode(outputPin,
OUTPUT); // define ultrasonic output pin myservo.attach(11); //
define the fifth pin of servo motor(PWM)
}




void advance(int a) // forward
{
digitalWrite(pinRB,LOW); // let motor act(right back) digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,LOW); // let motor act(left back)
digitalWrite(pinLF,HIGH); delay(a * 100); }
```

```
void right(int b) //turn right(single wheel)
{
digitalWrite(pinRB,LOW); //let motor act(right back)
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH); delay(b
* 100);
}


void left(int c) //turn left(single wheel)
{ digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,HIGH); digitalWrite(pinLB,LOW);
//let motor act(left back) digitalWrite(pinLF,HIGH);
delay(c * 100); }


void turnR(int d) //turn right(double wheel)
{
digitalWrite(pinRB,LOW); //let motor act(right back)
digitalWrite(pinRF,HIGH); digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,LOW); //let motor act(left front)
delay(d * 100); }


void turnL(int e) //turn left(double wheel)
{ digitalWrite(pinRB,HIGH); digitalWrite(pinRF,LOW);
//let motor act(right front) digitalWrite(pinLB,LOW);
//let motor act(left back) digitalWrite(pinLF,HIGH);
delay(e * 100); }


void stopp(int f) //stop
{ digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH);
delay(f * 100); }


void back(int g) //backward
{
digitalWrite(pinRB,HIGH); //let motor act(right back)
digitalWrite(pinRF,LOW); digitalWrite(pinLB,HIGH);
//let motor act(left back) digitalWrite(pinLF,LOW);
delay(g * 100); }


void ask_pin_F() // measure distance ahead
{
myservo.write(90);
digitalWrite(outputPin, LOW); // let ultrasonic transmit low voltage 2μs delayMicroseconds(2);
digitalWrite(outputPin, HIGH); // let ultrasonic transmit hight voltage 10μs, at least 10μs here
delayMicroseconds(10);
digitalWrite(outputPin, LOW); // keep ultrasonic transmitting low voltage
float Fdistance = pulseIn(inputPin, HIGH); // read time gap Fdistance=
Fdistance/5.8/10; // convert time into distance(unit:cm)
```

```
Serial.print("F distance:"); //output distance(unit:cm)
Serial.println(Fdistance); //display distance
Fspeedd = Fdistance; // read distance into Fspeedd(forward speed)
}


void ask_pin_L() // measure distance on the left side
{ myservo.write(5);
delay(delay_time);
digitalWrite(outputPin, LOW); // let ultrasonic transmit low voltage 2μs delayMicroseconds(2);
digitalWrite(outputPin, HIGH); // let ultrasonic transmit hight voltage 10μs, at least 10μs here
delayMicroseconds(10);
digitalWrite(outputPin, LOW); // keep ultrasonic transmitting low voltage
float Ldistance = pulseIn(inputPin, HIGH); // read time gap Ldistance=
Ldistance/5.8/10; // convert time into distance(unit:cm)
Serial.print("L distance:"); //output distance(unit:cm)
Serial.println(Ldistance); //display distance
Lspeedd = Ldistance; // read distance into Lspeedd(Leftward speed)
}


void ask_pin_R() // measure distance on the right side
{
myservo.write(177); delay(delay_time);
digitalWrite(outputPin, LOW); // let ultrasonic transmit low voltage 2μs delayMicroseconds(2);
digitalWrite(outputPin, HIGH); // let ultrasonic transmit high voltage 10μs, at least 10μs here delayMicroseconds(10);
digitalWrite(outputPin, LOW); // keep ultrasonic transmitting low voltage
float Rdistance = pulseIn(inputPin, HIGH); // read time gap Rdistance=
Rdistance/5.8/10; // convert time into distance(unit:cm)
Serial.print("R distance:"); //output distance(unit:cm)
Serial.println(Rdistance); //display distance
Rspeedd = Rdistance; // read distance into Rspeedd(rightward speed)
}


void detection() //measure three angles(0.90.179)
{
int delay_time = 250; // stabilizing time of servo motor after turning ask_pin_F(); //
read distance ahead
if(Fspeedd < 10) // if distance ahead is less than 10cm
{
stopp(1); // Clear output data back(2);
// backward for 0.2 sec
}
if(Fspeedd < 25) // if distance ahead is less than 25cm
{
stopp(1); // Clear output data ask_pin_L(); // read
distance on the left side delay(delay_time); // wait
for the servo to stabilize ask_pin_R(); // ead
distance on the right side delay(delay_time); // wait
for the servo to stabilize
if(Lspeedd > Rspeedd) //if the distance on the left side is larger than that of the right side
{
directionn = Rgo; //go rightwards
}
if(Lspeedd <= Rspeedd) //if the distance on the left side is no more than that of the right side
```

```
{
directionn = Lgo; //go leftwards
}
if (Lspeedd < 10 && Rspeedd < 10) //if the distance on both sides is less than 10cm
{
directionn = Bgo; //go backwards

} }
else //if the distance ahead if no less than 25cm
{
directionn = Fgo; //go forward
} }



void loop() {
myservo.write(90); //let servo motor return to its ready position, prepared for the next measurement
detection(); //measure angle and decide moving direction if(directionn == 2) //if directionn(direction) =
2(backward)
{
back(8); // backward(car)
turnL(2); //slightly move leftwards(prevent from being stuck in blind alley)
Serial.print(" Reverse "); //display direction(backward)
}
if(directionn == 6) //if directionn(direction) = 6(rightward)
{ back(1);
turnR(6); // turn right
Serial.print(" Right "); //display direction(turn left)
}
if(directionn == 4) //if directionn(direction) = 4(turn left)
{ back(1);
turnL(6); // turn left
Serial.print(" Left "); //display direction(turn right)
}
if(directionn == 8) //if directionn(direction) = 8(forward)
{
advance(1); // go forward as normal
Serial.print(" Advance "); //display direction(forward)
Serial.print(" ");
}
}
```