

```

#include <Servo.h>

byte sensorPinL = 6;
byte sensorPinR = 4;

byte sensorNorth = 7;
byte sensorEast = 8;
byte sensorWest = 13;
byte sensorSouth = 12;

int leftIR = 1;
int rightIR = 1;

int north = 0;
int east = 0;
int west = 0;
int south = 0;
int lastDirection = 0;
int currDirection = 0;
int neckDegrees = 85;
int steeringDegrees = 85;
int speedVal = 90;
int inches = 50;

const int trigPin = 2;
const int echoPin = 3;

// Motor setup
const int neckServo = 9;
const int steeringServo = 10;
const int driveServo = 11;

Servo neck;
Servo steering;
Servo drive;

void setup() {
  Serial.begin(9600);
  neck.attach(neckServo);
  steering.attach(steeringServo);
  drive.attach(driveServo);
  // neck.write(85);          // Neutral neck angle
  delay(50);
}

void loop() {

```

```

noseRead();
chooseSpeed();
findBeacon();
pingSensor();
printAll();
finalCheck();
}

int noseRead() {
  leftIR = digitalRead(sensorPinL);
  rightIR = digitalRead(sensorPinR);

  if (leftIR == 0 and rightIR == 0) {
  //  Serial.println("Dang! We're in a tight spot.");
  }

  if (leftIR == 0 and rightIR == 1) {
  //  Serial.println("Mermaid Off The Port Bow!");
  //  steeringDegrees += 5;
  }

  if (rightIR == 0 and leftIR == 1) {
  //  Serial.println("Right Obstacle Detected, Panic Then Turn Left");
  //  steeringDegrees -= 5;
  }

  if (leftIR == 1 and rightIR == 1) {
  //  steeringDegrees = 85;
  }
}

int printAll() {
  Serial.print("North: ");
  Serial.print(north);
  Serial.print(" East: ");
  Serial.print(east);
  Serial.print(" South: ");
  Serial.print(south);
  Serial.print(" West: ");
  Serial.print(west);
  Serial.print(" Left obstacle: ");
  Serial.print(leftIR);
  Serial.print(" Right obstacle: ");
  Serial.println(rightIR);
  Serial.print("Steering: ");
  Serial.print(steeringDegrees);
}

```

```

Serial.print(" Neck: ");
Serial.print(neckDegrees);
Serial.print(" Speed: ");
Serial.println(speedVal);
}

long microsecondsToInches(long microseconds) {
return microseconds / 74 / 2;
}

int findBeacon() {
north = digitalRead(sensorNorth);
east = digitalRead(sensorEast);
west = digitalRead(sensorWest);
south = digitalRead(sensorSouth);
// if (north == 0 and west == 0) {
// neckDegrees = 85;
// steeringDegrees = 85;
// return 0;
// }

if (north == 0) {
recordDirection();
currDirection = 0;
if (lastDirection == currDirection) {
if (neckDegrees > 70) {
neckDegrees -= 2;
}
if (neckDegrees <= 75) {
if (steeringDegrees >= 50) {
steeringDegrees -= 5;
}
}
}
}

if (west == 0) {
recordDirection();
currDirection = 90;
if (lastDirection == currDirection) {
if (neckDegrees < 100) {
neckDegrees += 2;
}
if (neckDegrees >= 95) {
if (steeringDegrees <= 130) {
steeringDegrees += 5;
}
}
}
}

```

```

    }
  }
}
if (east == 0) {
  recordDirection();
  currDirection = 270;
  if (lastDirection == currDirection) {
    if (neckDegrees > 70) {
      neckDegrees -= 1;
    }
  }
  // if (steeringDegrees >= 50) {
  //   steeringDegrees -= 5;
  // }
}
reverseSpeed();
}
if (south == 0) {
  recordDirection();
  currDirection = 180;
  if (lastDirection == currDirection) {
    if (neckDegrees < 100) {
      neckDegrees += 1;
    }
  }
  // if (steeringDegrees <= 130) {
  //   steeringDegrees += 5;
  // }
}
reverseSpeed();
}
}

```

```

int recordDirection() {
  lastDirection = currDirection;
  delay(50);
}

```

```

int chooseSpeed() {
  if (inches >= 45) {
    incrementSpeed();
  }
  if (inches <= 30) {
    decrementSpeed();
  }
  if (inches <= 15 and inches > 7) {
    stopSpeed();
  }
}

```

```

    if (inches < 7) {
        reverseSpeed();
    }
}

int pingSensor() {
    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);

    inches = microsecondsToInches(duration);
    Serial.print("Inches of clearance: ");
    Serial.println(inches);

    if (inches < 8) {
        Serial.println("OhgawdSTAAAAAAPPPP!!!");
    }
}

int writeAll() {
    drive.write(speedVal);
    neck.write(neckDegrees);
    steering.write(steeringDegrees);
}

int fullStop() {
    neckDegrees = 85;
    steeringDegrees = 85;
    speedVal = 95;
}

int incrementSpeed() {
    if (speedVal < 110) {
        speedVal += 3;
    }
}

int decrementSpeed() {
    if (speedVal > 95) {
        speedVal -= 3;
    }
}

```

```

}

int stopSpeed() {
    speedVal = 95;
}

int reverseSpeed() {
    if (speedVal > 55) {
        speedVal -= 1
    }
}

int finalCheck() {
    if (speedVal < 85) {
        steeringDegrees = 170 - neckDegrees;    // Reverse steering
    }
    if (speedVal > 85 and speedVal <= 95) {
        steeringDegrees = 85;
    }
    if (leftIR == 1 and rightIR == 1) {
        writeAll();           // No obstacles
    }
    if (leftIR == 0 or rightIR == 0) {
        drive.write(95);      // Obstacles present
    }
    if (north == 1 and south == 1 and west == 1 and east == 1) {
        speedVal = 95;
    }
}

```