// TestCode for ultrasonic Sensor HC-SR04 https://www.amazon.com/gp/product/B01COSN7O6/

// D. Brancazio 7/2/2018

// Does a ping about 20 times per second and displays distance read

// ---------------------------------------------------------------------------

// OTHER ARDUINO AND MAIN PARTS,

//FIX CALCULATIONS FOR DELAY, ask about diastance tracker but it on adafruit

#include <NewPing.h> // library for sensor

#include <Adafruit\_NeoPixel.h>

#include <avr/power.h>

#define PIN 7

#define TRIGGER\_PIN 2 // Arduino pin tied to trigger pin on the ultrasonic sensor

#define ECHO\_PIN 3 // Arduino pin tied to echo pin on the ultrasonic sensor.

#define MAX\_DISTANCE 200 // Maximum distance we want to ping for (in cm). Sensor rating is 400-500cm max distance

#define NUMPIXELS 6

#define enableL 9 // ENA of L298n

#define LmotorPin1 8 // IN1 of L298n

#define LmotorPin2 11 // IN2 of L298n

//Motor B - Left

#define enableR 10 // ENb of L298n

#define RmotorPin1 12 // IN3 of L298n

#define RmotorPin2 13 // IN4 of L298n

#define FORWARDSPEED 150

#define BACKWARDSPEED 150

NewPing sensor1(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANCE); // declare sensor1 as defined in NewPing library

Adafruit\_NeoPixel pixels = Adafruit\_NeoPixel(NUMPIXELS, PIN, NEO\_GRB + NEO\_KHZ800);

#include <Servo.h> // servo library from Arduino

# define servoPin 4

Servo Servo1; // define Servo name

//end position for the servo

int leftDIS;

int choice;

int direction;

long randNumber;

long randNumber2;

long randNumber3;

int delayTime = 1000; //time between commands

int Distance;

int distanceL;

int distanceR;

int pos;

int Direction;

int distance;

int startPos=0;

int middlePos=90;

int endPos=180;

//b able to have user put in distance go that distance around it and get obejct and maybe back use delay for the distance of user divided by speed of motor

void setup()

{

Serial.begin(115200); // Open serial monitor at 115200 baud to see ping results.

randomSeed(analogRead(0));

Servo1.attach(servoPin); // set up servos on digital pins 8

#if defined (\_\_AVR\_ATtiny85\_\_)

if (F\_CPU == 16000000) clock\_prescale\_set(clock\_div\_1);

#endif

pixels.begin(); // This initializes the NeoPixel library.

//Set pins as outputs

pinMode(enableL, OUTPUT);

pinMode(LmotorPin1, OUTPUT);

pinMode(LmotorPin2, OUTPUT);

pinMode(enableR, OUTPUT);

pinMode(RmotorPin1, OUTPUT);

pinMode(RmotorPin2, OUTPUT);

}

//user is gonna put in small distance, gonna turn to that position and then it will sense in that direction, and if it finds somsthing within that range it will (5cm off), go forward

//try to add in obstacles later

void loop() //left is 180 and right is 0 for pos

{

randNumber = random(0, 100);

randNumber2 = random(0, 100);

randNumber3 = random(0, 100);

Serial.print("What direction do you want to go in?");

while(Serial.available() == 0) {}

Direction=Serial.parseInt();

digitalWrite(enableR, HIGH);

digitalWrite(enableL, HIGH);

if(Direction==1)//right

{

goright();

}

if(Direction==2)//left

{

goleft();

}

if(Direction==4)//right down/south

{

gorightdown();

}//forward

pos = middlePos;

if(pos==90)

{

Servo1.write(pos); // go to specified position in degrees

delay(1000);

Serial.println(sonarsearch(Distance));

if ((sonarsearch(Distance)) == 0 || (sonarsearch(Distance)) > 15)//no object

{

for (int i = 0; i <= NUMPIXELS; i++) {

pixels.setPixelColor(i, pixels.Color(randNumber, randNumber2, randNumber3)); // Moderately bright green color.//changing colors when moving

pixels.show();

}

}

else

{

for (int i = 0; i <= NUMPIXELS; i++) {

pixels.setPixelColor(i, pixels.Color((150 - 10 \* (sonarsearch(Distance))), 10 \* (sonarsearch(Distance)) - 10, 0)); // Moderately bright green color

pixels.show();

}

pos = startPos; //for left 0

Servo1.write(pos);

delay(1000);

}

}

if (pos == 0) //This is for the left side

{

delay(1000);

Serial.print(sonarsearch(distanceL));

if ((sonarsearch(distanceL)) == 0 || (sonarsearch(distanceL)) > 15)//no object servo goes to 90 and move left

{

leftDIS= sonarsearch(distanceL);

pos = 90;

Servo1.write(pos);

delay(1000);

directionturn();

for (int i = 0; i <= NUMPIXELS; i++) {

pixels.setPixelColor(i, pixels.Color(0, 0, 0)); //lights go off for LEFT TURN

pixels.show();

delay(200);

}

}

else//servo goes to 180 and checks

{

leftDIS= sonarsearch(distanceL);

pos = 180;

Servo1.write(pos);

delay(1000);

for (int i = 0; i <= NUMPIXELS; i++) {

pixels.setPixelColor(i, pixels.Color(100, 0, 0)); //lights go off for RIGHT TURN

pixels.show();

delay(200);

}

}

}

if(pos==180)//This is for the right side

{

delay(1000);

Serial.print(sonarsearch(distanceR));

if((sonarsearch(distanceR))==0||(sonarsearch(distanceR))>15)//no object servo at 90 and move right

{

pos=90;

Servo1.write(pos);

delay(1000);

//lights for right turn ON

directionturn();

}

else//if object

{

if(leftDIS>(sonarsearch(distanceR)))//turn left

{

pos=0;

Servo1.write(pos);

delay(1000);

//turn on lights for LEFT

directionturn();

}

else//turn right

{

pos=90;

Servo1.write(pos);

delay(1000);

//turn lights on for RIGHT

directionturn();

}

}

}

}

int sonarsearch(int distance)

{

delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay between pings.

distance = sensor1.ping\_cm(); // Send ping, get ping time in microseconds (uS).

return distance;

}

void directionturn()

{

digitalWrite(enableR, HIGH);

digitalWrite(enableL, HIGH);

digitalWrite(LmotorPin1, LOW);//stop for three seconds

digitalWrite(LmotorPin2, LOW);

digitalWrite(RmotorPin1, LOW);

digitalWrite(RmotorPin2, LOW);

delay(3000);

if ((leftDIS == 0 || leftDIS > 15)||(leftDIS>(sonarsearch(distanceR))))

{

digitalWrite(LmotorPin1, LOW);//forward left

digitalWrite(LmotorPin2, HIGH);

delay(2000);

digitalWrite(LmotorPin2, LOW);

}

else

{

digitalWrite(RmotorPin1, LOW);//forward right

digitalWrite(RmotorPin2, HIGH);

delay(2000);

digitalWrite(RmotorPin2, LOW);

}

digitalWrite(RmotorPin1, HIGH);//straight

digitalWrite(RmotorPin2, LOW);

digitalWrite(LmotorPin2, HIGH);

digitalWrite(LmotorPin1, LOW);

delay(3000);

if ((leftDIS == 0 || leftDIS > 15)||(leftDIS>(sonarsearch(distanceR))))

{

digitalWrite(RmotorPin1, LOW);//forward right

digitalWrite(RmotorPin2, HIGH);

delay(2000);

digitalWrite(RmotorPin2, LOW);

}

else

{

digitalWrite(LmotorPin1, LOW);//forward left

digitalWrite(LmotorPin2, HIGH);

delay(2000);

digitalWrite(LmotorPin2, LOW);

}

digitalWrite(RmotorPin1, HIGH);//straight

digitalWrite(RmotorPin2, LOW);

digitalWrite(LmotorPin2, HIGH);

digitalWrite(LmotorPin1, LOW);

}

void goright(){

analogWrite(RmotorPin1, 150);

analogWrite(RmotorPin2, 0);

delay(2000);

analogWrite(RmotorPin1, 0);

}

void goleft()

{

analogWrite(LmotorPin1, 0);

analogWrite(LmotorPin2, 150);

delay(6000);

analogWrite(LmotorPin2, 0);

}

void gorightdown()

{

analogWrite(RmotorPin1, 0);

analogWrite(RmotorPin2, 150);

delay(5000);

analogWrite(RmotorPin2, 0);

}