#include <Wire.h>

#include <Adafruit\_MotorShield.h>

#include "utility/Adafruit\_MS\_PWMServoDriver.h"

#include <Servo.h>

// Create the motor shield object with the default I2C address

Adafruit\_MotorShield AFMS = Adafruit\_MotorShield();

//define motor variables

Adafruit\_DCMotor \*MovingMotor = AFMS.getMotor(4); //Motor M4

Adafruit\_DCMotor \*TiltingMotor = AFMS.getMotor(1); //Motor M1

// define global variables

#define echoPin 2 // Echo Pin

#define trigPin 3 // Trigger Pin

int setspeed = 55; // sets global speed which changed depending on if it is turning or going straight

int state = 0; // 2=backward,1=forward

int i;

int pos = 0;

int maximumRange = 45; // Maximum range needed

long duration, distance; // Duration used to calculate distance

Servo servoupdown;

Servo servorightleft;

int greenpin = 5; //select the pin for the red LED

int redpin = 6;// select the pin for the green LED

int bluepin =11; // select the pin for the blue LED

unsigned long previousMillis = 0; // will store last time LED was updated

const long interval = 1000; // interval at which to blink (milliseconds)

int ledState = LOW; // ledState used to set the LED

void setup() {

Serial.begin(115200); // set up Serial library at 9600 bps

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

servoupdown.attach(10);

servorightleft.attach(9);

servoupdown.write(0);

servorightleft.write(113);

Serial.println("Project RC Car - Motor Test");

AFMS.begin(); // create with the default frequency 1.6KHz

MovingMotor->setSpeed(setspeed); //Sets speed for motor

MovingMotor->run(FORWARD);

// turn on motor

MovingMotor->run(RELEASE);

TiltingMotor->run(RELEASE);

pinMode(redpin, OUTPUT);

pinMode(greenpin, OUTPUT);

pinMode(bluepin, OUTPUT);

}

// If something in front,reverse to any direction and the LED light will blink

// While reversing, if something block it's way, it will go forward to any direction.

void loop() {

servomove(); // Calls the function to move the tilting servos (also scans its surroundings)

switch (state){

case 1:

setcolor(0, 50, 0);

forward();

break;

case 2:

setcolor(50, 0, 0);

backward();

break;

default:

state = 1;

break;

}

}

void servomove(){

servoupdown.write(40); // tell servo to go to position in variable 'pos'

for(pos = 113; pos < 160; pos += 1) // goes from 0 degrees to 180 degrees

{ // in steps of 1 degree

servorightleft.write(pos); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

senseobject();

}

for(pos = 160; pos>=66; pos-=1) // goes from 180 degrees to 0 degrees

{

servorightleft.write(pos); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

senseobject();

}

for(pos = 66; pos<113; pos +=1)

{

servorightleft.write(pos);

delay(15);

senseobject();

}

}

void randomturn(){

//random function gives a random number from (MIN, MAX-1)

int turn = random(0,4); // 0=right, 1=left, 2=stright

if(turn == 0){

setspeed = 85;

right();

Serial.println(turn);

}

else if(turn == 1){

setspeed = 85;

left();

Serial.println(turn);

}

else{

setspeed = 50;

straight();

Serial.println(turn);

}

}

void senseobject(){

/\* The following trigPin/echoPin cycle is used to determine the

distance of the nearest object by bouncing soundwaves off of it. \*/

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(8);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

//Calculate the distance (in cm) based on the speed of sound.

distance = duration/58.2;

if (distance == 0){

Serial.println("distance is 0, going back");

state = 2;

}

if (distance >= maximumRange){

Serial.println("No Objects in front, going forward");

Serial.println(distance);

state = 1;

}

else {

Serial.println("found object, going back");

Serial.println(distance);

state = 2;

}

}

void backward() {

randomturn();

MovingMotor->run(BACKWARD);

for (i=0; i<setspeed; i++){

MovingMotor->setSpeed(i);

//delay(10);

}

}

void forward() {

randomturn();

MovingMotor->run(FORWARD);

MovingMotor->setSpeed(setspeed);

//delay(10);

}

void slowdown(){

for (i=setspeed; i>0; i--) {

MovingMotor->setSpeed(i);

//delay(10);

}

}

void stoped() {

MovingMotor->run(RELEASE);

//delay(500);

}

void left() {

TiltingMotor->run(FORWARD);

TiltingMotor->setSpeed(255);

//delay(60);

}

void right() {

TiltingMotor->run(BACKWARD);

TiltingMotor->setSpeed(255);

//delay(60);

}

void straight(){

TiltingMotor->run(RELEASE);

//delay(50);

}

void setcolor(int red, int green, int blue){

analogWrite(redpin, red);

analogWrite(greenpin, green);

analogWrite(bluepin, blue);

delay(10); }