#define leftCenterSensor A3

#define leftNearSensor A4

#define leftFarSensor A5

#define rightCenterSensor A2

#define rightNearSensor A1

#define rightFarSensor A0

#define leapTime 200

#define leftMotor1 9

#define leftMotor2 8

#define rightMotor1 7

#define rightMotor2 6

#define led 13

int leftCenterReading;

int leftNearReading;

int leftFarReading;

int rightCenterReading;

int rightNearReading;

int rightFarReading;

int leftNudge;

int replaystage;

int rightNudge;

char path[30] = {};

int pathLength;

int readLength;

void setup(){

pinMode(leftCenterSensor, INPUT);

pinMode(leftNearSensor, INPUT);

pinMode(leftFarSensor, INPUT);

pinMode(rightCenterSensor, INPUT);

pinMode(rightNearSensor, INPUT);

pinMode(rightFarSensor, INPUT);

pinMode(leftMotor1, OUTPUT);

pinMode(leftMotor2, OUTPUT);

pinMode(rightMotor1, OUTPUT);

pinMode(rightMotor2, OUTPUT);

pinMode(led, OUTPUT);

//Serial.begin(115200);

digitalWrite(led, HIGH);

delay(1000);

}

void loop(){

readSensors();

if(leftFarReading<200 && rightFarReading<200 &&

(leftCenterReading>200 || rightCenterReading>200) ){

straight();

}

else{

leftHandWall();

}

}

void readSensors(){

leftCenterReading = analogRead(leftCenterSensor);

leftNearReading = analogRead(leftNearSensor);

leftFarReading = analogRead(leftFarSensor);

rightCenterReading = analogRead(rightCenterSensor);

rightNearReading = analogRead(rightNearSensor);

rightFarReading = analogRead(rightFarSensor);

// serial printing below for debugging purposes

// Serial.print("leftCenterReading: ");

// Serial.println(leftCenterReading);

// Serial.print("leftNearReading: ");

// Serial.println(leftNearReading);

// Serial.print("leftFarReading: ");

// Serial.println(leftFarReading);

// Serial.print("rightCenterReading: ");

// Serial.println(rightCenterReading);

// Serial.print("rightNearReading: ");

// Serial.println(rightNearReading);

// Serial.print("rightFarReading: ");

// Serial.println(rightFarReading);

// delay(200);

}

void leftHandWall(){

if( leftFarReading>200 && rightFarReading>200){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(leapTime);

readSensors();

if(leftFarReading>200 || rightFarReading>200){

done();

}

if(leftFarReading<200 && rightFarReading<200){

turnLeft();

}

}

if(leftFarReading>200){ // if you can turn left then turn left

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(leapTime);

readSensors();

if(leftFarReading<200 && rightFarReading<200){

turnLeft();

}

else{

done();

}

}

if(rightFarReading>200){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(30);

readSensors();

if(leftFarReading>200){

delay(leapTime-30);

readSensors();

if(rightFarReading>200 && leftFarReading>200){

done();

}

else{

turnLeft();

return;

}

}

delay(leapTime-30);

readSensors();

if(leftFarReading<200 && leftCenterReading<200 &&

rightCenterReading<200 && rightFarReading<200){

turnRight();

return;

}

path[pathLength]='S';

// Serial.println("s");

pathLength++;

//Serial.print("Path length: ");

//Serial.println(pathLength);

if(path[pathLength-2]=='B'){

//Serial.println("shortening path");

shortPath();

}

straight();

}

readSensors();

if(leftFarReading<200 && leftCenterReading<200 && rightCenterReading<200

&& rightFarReading<200 && leftNearReading<200 && rightNearReading<200){

turnAround();

}

}

void done(){

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

replaystage=1;

path[pathLength]='D';

pathLength++;

while(analogRead(leftFarSensor)>200){

digitalWrite(led, LOW);

delay(150);

digitalWrite(led, HIGH);

delay(150);

}

delay(500);

replay();

}

void turnLeft(){

while(analogRead(rightCenterSensor)>200||analogRead(leftCenterSensor)>200){

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, HIGH);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(2);

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(1);

}

while(analogRead(rightCenterSensor)<200){

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, HIGH);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(2);

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(1);

}

if(replaystage==0){

path[pathLength]='L';

//Serial.println("l");

pathLength++;

//Serial.print("Path length: ");

//Serial.println(pathLength);

if(path[pathLength-2]=='B'){

//Serial.println("shortening path");

shortPath();

}

}

}

void turnRight(){

while(analogRead(rightCenterSensor)>200){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, HIGH);

delay(2);

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(1);

}

while(analogRead(rightCenterSensor)<200){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, HIGH);

delay(2);

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(1);

}

while(analogRead(leftCenterSensor)<200){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, HIGH);

delay(2);

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(1);

}

if(replaystage==0){

path[pathLength]='R';

Serial.println("r");

pathLength++;

Serial.print("Path length: ");

Serial.println(pathLength);

if(path[pathLength-2]=='B'){

Serial.println("shortening path");

shortPath();

}

}

}

void straight(){

if( analogRead(leftCenterSensor)<200){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(1);

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(5);

return;

}

if(analogRead(rightCenterSensor)<200){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(1);

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(5);

return;

}

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(4);

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(1);

}

void turnAround(){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(150);

while(analogRead(leftCenterSensor)<200){

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, HIGH);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(2);

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(1);

}

path[pathLength]='B';

pathLength++;

straight();

//Serial.println("b");

//Serial.print("Path length: ");

//Serial.println(pathLength);

}

void shortPath(){

int shortDone=0;

if(path[pathLength-3]=='L' && path[pathLength-1]=='R'){

pathLength-=3;

path[pathLength]='B';

//Serial.println("test1");

shortDone=1;

}

if(path[pathLength-3]=='L' && path[pathLength-1]=='S' && shortDone==0){

pathLength-=3;

path[pathLength]='R';

//Serial.println("test2");

shortDone=1;

}

if(path[pathLength-3]=='R' && path[pathLength-1]=='L' && shortDone==0){

pathLength-=3;

path[pathLength]='B';

//Serial.println("test3");

shortDone=1;

}

if(path[pathLength-3]=='S' && path[pathLength-1]=='L' && shortDone==0){

pathLength-=3;

path[pathLength]='R';

//Serial.println("test4");

shortDone=1;

}

if(path[pathLength-3]=='S' && path[pathLength-1]=='S' && shortDone==0){

pathLength-=3;

path[pathLength]='B';

//Serial.println("test5");

shortDone=1;

}

if(path[pathLength-3]=='L' && path[pathLength-1]=='L' && shortDone==0){

pathLength-=3;

path[pathLength]='S';

//Serial.println("test6");

shortDone=1;

}

path[pathLength+1]='D';

path[pathLength+2]='D';

pathLength++;

//Serial.print("Path length: ");

//Serial.println(pathLength);

//printPath();

}

void printPath(){

Serial.println("+++++++++++++++++");

int x;

while(x<=pathLength){

Serial.println(path[x]);

x++;

}

Serial.println("+++++++++++++++++");

}

void replay(){

readSensors();

if(leftFarReading<200 && rightFarReading<200){

straight();

}

else{

if(path[readLength]=='D'){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(100);

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

endMotion();

}

if(path[readLength]=='L'){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(leapTime);

turnLeft();

}

if(path[readLength]=='R'){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(leapTime);

turnRight();

}

if(path[readLength]=='S'){

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(leapTime);

straight();

}

readLength++;

}

replay();

}

void endMotion(){

digitalWrite(led, LOW);

delay(500);

digitalWrite(led, HIGH);

delay(200);

digitalWrite(led, LOW);

delay(200);

digitalWrite(led, HIGH);

delay(500);

endMotion();

}

