**8TH WEEK’S REPORT**

Maze Solving Robot

1. Code for optimization

This code can be compiled well without error, but I have tested if it really works as I do not have the Arduino board.

char exppath[100]="SLBRSRLSSRR";//define the path

int pathlength=10;

int xcurrenttile=0;

void setup()

{

Serial.begin(115200);

}

void loop()

{

while(xcurrenttile<pathlength)//strat searching

{

if (exppath[xcurrenttile]=='B')//find the character B

{

if (exppath[xcurrenttile-1]=='L'&&exppath[xcurrenttile+1]=='L')

{

exppath[xcurrenttile-1]='S';

}

else if (exppath[xcurrenttile-1]=='L'&&exppath[xcurrenttile+1]=='S')

{

exppath[xcurrenttile-1]='R';

}

else if (exppath[xcurrenttile-1]=='L'&&exppath[xcurrenttile+1]=='R')

{

exppath[xcurrenttile-1]='B';

}

else if (exppath[xcurrenttile-1]=='R'&&exppath[xcurrenttile+1]=='L')

{

exppath[xcurrenttile-1]='B';

}

else if (exppath[xcurrenttile-1]=='R'&&exppath[xcurrenttile+1]=='R')

{

exppath[xcurrenttile-1]='S';

}

else if (exppath[xcurrenttile-1]=='R'&&exppath[xcurrenttile+1]=='S')

{

exppath[xcurrenttile-1]='L';

}

else if (exppath[xcurrenttile-1]=='S'&&exppath[xcurrenttile+1]=='S')

{

exppath[xcurrenttile-1]='B';

}

else if (exppath[xcurrenttile-1]=='S'&&exppath[xcurrenttile+1]=='R')

{

exppath[xcurrenttile-1]='L';

}

else if (exppath[xcurrenttile-1]=='S'&&exppath[xcurrenttile+1]=='L')

{

exppath[xcurrenttile-1]='R';

}

for (xcurrenttile;xcurrenttile<pathlength-2;xcurrenttile++)//reduce the path size by 2

{

exppath[xcurrenttile]=exppath[xcurrenttile+2];

}

xcurrenttile--;

pathlength=pathlength-2;

Serial.println("Optimized one is:");

Serial.println(exppath);

delay (300);

}

else

{

Serial.println("Optimized one is:");

Serial.println(exppath);

delay (300);

xcurrenttile++;

}

}

}

1. Sensor reading part

As we have 7 sensors in our robot, I define Sensor A~G in this code.

#define SensorA 1

#define SensorB 2

#define SensorC 3

#define SensorD 4

#define SensorE 5

#define SensorF 6

#define SensorG 7

int SensorAReading;

int SensorBReading;

int SensorCReading;

int SensorDReading;

int SensorEReading;

int SensorFReading;

int SensorGReading;

void setup(){

pinMode(SensorA, INPUT);

pinMode(SensorB, INPUT);

pinMode(SensorC, INPUT);

pinMode(SensorD, INPUT);

pinMode(SensorE, INPUT);

pinMode(SensorF, INPUT);

pinMode(SensorF, INPUT);

}

void loop(){

readSensors();

if(SensorAReading>200){

digitalWrite(1, HIGH);

}

else{

digitalWrite(1, LOW);

}

if(SensorBReading>200){

digitalWrite(2, HIGH);

}

else{

digitalWrite(2, LOW);

}

if(SensorCReading>200){

digitalWrite(3, HIGH);

}

else{

digitalWrite(3, LOW);

}

if(SensorDReading>200){

digitalWrite(4, HIGH);

}

else{

digitalWrite(4, LOW);

}

if(SensorEReading>200){

digitalWrite(5, HIGH);

}

else{

digitalWrite(5, LOW);

}

if(SensorFReading>200){

digitalWrite(6, HIGH);

}

else{

digitalWrite(6, LOW);

}

if(SensorGReading>200){

digitalWrite(7, HIGH);

}

else{

digitalWrite(7, LOW);

}

}

void readSensors(){

SensorAReading = analogRead(SensorA);

SensorBReading = analogRead(SensorB);

SensorCReading = analogRead(SensorC);

SensorDReading = analogRead(SensorD);

SensorEReading = analogRead(SensorE);

SensorFReading = analogRead(SensorF);

SensorGReading = analogRead(SensorG); }

1. Motor test part

This code can test if the robot can do turn left/right, stop and turn around.

#define SensorA 1

#define SensorB 2

#define SensorC 3

#define SensorD 4

#define SensorE 5

#define SensorF 6

#define SensorG 7

int SensorAReading;

int SensorBReading;

int SensorCReading;

int SensorDReading;

int SensorEReading;

int SensorFReading;

int SensorGReading;

#define leftMotor1 8

#define leftMotor2 9

#define rightMotor1 10

#define rightMotor2 11

void setup(){

pinMode(SensorA, INPUT);

pinMode(SensorB, INPUT);

pinMode(SensorC, INPUT);

pinMode(SensorD, INPUT);

pinMode(SensorE, INPUT);

pinMode(SensorF, INPUT);

pinMode(SensorF, 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()

{

//Moves straight for 2.5 seconds

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(2500);

//STOPS for 1 second

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(10000);

//Turns right(1.075 seconds)

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(1075);

//Straight for 1 seconds

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(1000);

//Turns left(1.075 seconds)

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(1075);

//Straight for 1 seconds

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(1000);

//Turns around 180 degrees

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, HIGH);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

delay(900);

//Stop for 1 second

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

delay(10000);

}