**ROBO WAITER CODE**

///// Robo waiter program //////

#define motor\_11 2

#define motor\_12 3

#define motor\_21 4

#define motor\_22 5

#define front\_sensor 6

#define sw1 7

//#define sw2 8

#define left\_line 9

#define right\_line 10

unsigned char i=0,count=0,z=0;

int buttonState = 0;

void setup(){

Serial.begin(9600);

pinMode(motor\_11,OUTPUT);

pinMode(motor\_12,OUTPUT);

pinMode(motor\_21,OUTPUT);

pinMode(motor\_22,OUTPUT);

pinMode(left\_line,INPUT);

pinMode(right\_line,INPUT);

pinMode(front\_sensor,INPUT);

pinMode(sw1,INPUT\_PULLUP);

// pinMode(sw2,INPUT\_PULLUP);

}

//\*\*\*\*\*functions to run motors\*\*\*\*\*//

void forward()

{

digitalWrite(motor\_11,HIGH);

digitalWrite(motor\_12,LOW);

digitalWrite(motor\_21,LOW);

digitalWrite(motor\_22,HIGH);

}

void backward()

{

digitalWrite(motor\_11,LOW);

digitalWrite(motor\_12,HIGH);

digitalWrite(motor\_21,HIGH);

digitalWrite(motor\_22,LOW);

}

void right()

{

digitalWrite(motor\_11,LOW);

digitalWrite(motor\_12,LOW);

digitalWrite(motor\_21,LOW);

digitalWrite(motor\_22,HIGH);

}

void hard\_right()

{

digitalWrite(motor\_11,LOW);

digitalWrite(motor\_12,HIGH);

digitalWrite(motor\_21,LOW);

digitalWrite(motor\_22,HIGH);

}

void left()

{

digitalWrite(motor\_11,HIGH);

digitalWrite(motor\_12,LOW);

digitalWrite(motor\_21,LOW);

digitalWrite(motor\_22,LOW);

}

void hard\_left()

{

digitalWrite(motor\_11,HIGH);

digitalWrite(motor\_12,LOW);

digitalWrite(motor\_21,HIGH);

digitalWrite(motor\_22,LOW);

}

void Break()

{

digitalWrite(motor\_11,LOW);

digitalWrite(motor\_12,LOW);

digitalWrite(motor\_21,LOW);

digitalWrite(motor\_22,LOW);

}

void loop(){

buttonState = digitalRead(sw1);

///////////////////Table - 1 //////////////////

if(buttonState==LOW){

Serial.println("1st route");

if(digitalRead(front\_sensor)==true){

Break();

}

if((digitalRead(left\_line)==false) && (digitalRead(right\_line)==false)&&i==0){

i=1;

}

else if((digitalRead(left\_line)==true) && (digitalRead(right\_line)==true)&&i==1){

count = count+1;

Serial.println(count);

i=0;

z=1;

delay(50);

}

//////////////////////////////////////////////////////////////////////////////////////////

else if((count == 1)&&(z==1)){

Serial.println("1st intersection");

Break();

delay(2000);

left();

while(digitalRead(left\_line)==true);

z=0;

}

else if((count == 2)&&(z==1)){

Serial.println("2nd intersection");

Break();

delay(2000);

hard\_right();

delay(300);

right();

while(digitalRead(right\_line)==true);

z=0;

}

else if((count == 3)&&(z==1)){

Serial.println("3rd intersection");

Break();

delay(1500);

left();

while(digitalRead(left\_line)==true);

z=0;

}

else if((count == 4)&&(z==1)){

Serial.println("4th intersection");

Break();

delay(1500);

right();

while(digitalRead(right\_line)==true);

z=0;

}

else if((count == 6)&&(z==1)){

Serial.println("5th intersection");

while(1){

Break();

}

}

//////////////////////////////LINE Tracking/////////////////////////////////////////////////

else if(digitalRead(left\_line)==false){

left();

// while(digitalRead(left\_line)==false);

}

else if(digitalRead(right\_line)==false){

right();

// while(digitalRead(right\_line)==false);

}

else{

forward();

}

}

/////////////////Table - 2///////////////////////////////

else if(buttonState==HIGH){

// Serial.println("2nd route");

if(digitalRead(front\_sensor)==true){

Break();

}

if((digitalRead(left\_line)==false) && (digitalRead(right\_line)==false)&&i==0){

i=1;

}

else if((digitalRead(left\_line)==true) && (digitalRead(right\_line)==true)&&i==1){

count = count+1;

Serial.println(count);

i=0;

z=1;

}

//////////////////////////////////////////////////////////////////////////////////////////

else if((count == 2)&&(z==1)){

Serial.println("2nd intersection");

Break();

delay(2000);

left();

while(digitalRead(left\_line)==true);

z=0;

}

else if((count == 3)&&(z==1)){

Serial.println("2nd intersection");

Break();

delay(2000);

hard\_right();

delay(500);

right();

while(digitalRead(right\_line)==true);

z=0;

}

else if((count == 4)&&(z==1)){

Serial.println("3rd intersection");

Break();

delay(1500);

left();

while(digitalRead(left\_line)==true);

z=0;

}

else if((count == 5)&&(z==1)){

Serial.println("4th intersection");

Break();

delay(1500);

right();

while(digitalRead(right\_line)==true);

z=0;

}

else if((count == 7)&&(z==1)){

Serial.println("5th intersection");

while(1){

Break();

}

}

///////////////////////////LINE Tracking////////////////////////////////////////////////

else if(digitalRead(left\_line)==false){

left();

// while(digitalRead(left\_line)==false);

}

else if(digitalRead(right\_line)==false){

right();

// while(digitalRead(right\_line)==false);

}

else{

forward();

}

}

}