

Rapport de la séance 6

16 janvier 2024

WARTSKI NARANJO Daniel

Robotique

Travail réalisé

Lors de cette séance, j'ai décidé de modifier mon programme afin que mon robot puisse suivre la ligne noire de manière optimale. À cet effet, j'ai introduit de nouvelles fonctions telles que `rotate_right`, `rotate_left`, `custom_turn`, `little_left`, `little_right`, et `suiv_ligne`. De plus, j'ai apporté des modifications aux fonctions `left` et `right`. L'algorithme permettant au robot de détecter la ligne noire demeure inchangé.

```
1  #define M1A 11
2  #define M1B 9
3  #define M2A 6
4  #define M2B 5
5  #define IR1 2
6  #define IR2 3
7  #define IR3 4
8  #define echo 8
9  #define trig 12
10 int distance;
11 int lecture_echo;
12 int v=200;
13
14 String maMatrice[100]; // Déclaration d'un tableau de chaînes de caractères avec une taille maximale de 100
15 int n = 0; // Initialisation de la variable n à 0
16
17 void setup() {
18     // put your setup code here, to run once:
19     Serial.begin(9550);
20     pinMode(M1A,OUTPUT);
21     pinMode(M2A,OUTPUT);
22     pinMode(M1B,OUTPUT);
23     pinMode(M2B,OUTPUT);
24     pinMode(IR1,INPUT);
25     pinMode(IR2,INPUT);
26     pinMode(IR3,INPUT);
27     pinMode(echo,INPUT);
28     pinMode(trig,OUTPUT);
29
30     maMatrice[n]="Debut";
31     delay(1000);
32     stop();
33 }
```

```

35 void loop() {
36   // put your main code here, to run repeatedly:
37   suiv_ligne();
38 }
39 void forward(){
40   analogWrite(M1B,0);
41   analogWrite(M2B,0);
42   analogWrite(M1A,v);
43   analogWrite(M2A,v);
44 }
45 void backward(){
46   analogWrite(M1A,0);
47   analogWrite(M2A,0);
48   analogWrite(M1B,v);
49   analogWrite(M2B,v);
50 }
51 void rotate_right(){
52   analogWrite(M1B,0);
53   analogWrite(M2A,0);
54   analogWrite(M1A,v*0.75);
55   analogWrite(M2B,v*0.75);
56 }
57 void rotate_left(){
58   analogWrite(M1A,0);
59   analogWrite(M2B,0);
60   analogWrite(M1B,v*0.75);
61   analogWrite(M2A,v*0.75);
62 }

```

```

63 void stop(){
64   analogWrite(M1A,0);
65   analogWrite(M2A,0);
66   analogWrite(M1B,0);
67   analogWrite(M2B,0);
68 }
69 void custom_turn(float RightSPEED, float LeftSpeed){
70   analogWrite(M1B,0);
71   analogWrite(M2B,0);
72   analogWrite(M2A, v*RightSPEED);
73   analogWrite(M1A, v*LeftSpeed);
74 }
75 void right(){
76   custom_turn(0.5,1);
77 }
78 void left(){
79   custom_turn(1,0.5);
80 }
81 void little_right(){
82   custom_turn(0.9,1);
83 }
84 void little_left(){
85   custom_turn(1,0.9);
86 }

```

```

87 void suiv_ligne(){
88   int IRC=digitalRead(IR2);
89   int IRR=digitalRead(IR1);
90   int IRL=digitalRead(IR3);
91   if (IRC==0){
92     if (IRR==1 && IRL==0){
93       little_left();
94     }
95     else if (IRR==0 && IRL==1){
96       little_right();
97     }
98     else if (IRR==0 && IRL==0){
99       stop();
100     }
101     else if (IRR==1 && IRL==1){
102       forward();
103     }
104   }
105   else if (IRC==1){
106     if (IRR==1 && IRL==0){
107       left();
108       n=n+1;
109       maMatrice[n]="l";
110     }
111     else if (IRR==0 && IRL==1){
112       right();
113       n=n+1;
114       maMatrice[n]="r";
115     }

```

```

116   else if (IRR==0 && IRL==0){
117     if (random(2) == 0) {
118       right();
119       n=n+1;
120       maMatrice[n]="r";
121     }
122     else {
123       left();
124       n=n+1;
125       maMatrice[n]="l";
126     }
127   }
128   }
129   else if (IRR==1 && IRL==1){
130     if(n>0 && maMatrice[n-1]=="r"){
131       unsigned long tempsDebut=millis();
132       while(millis() - tempsDebut < 1000){
133         rotate_right();
134       }
135     }
136     else if(n>0 && maMatrice[n-1]=="l"){
137       unsigned long tempsDebut=millis();
138       while(millis() - tempsDebut < 1000){
139         rotate_left();
140       }
141     }
142   }
143 }
144 }

```

Objectifs pour la prochaine séance

- Concevoir le modèle 3D du support pour le capteur ultrason.
- Poursuivre le développement du programme du robot.