

Séance du 08/01/2024 Planning : Mise en place du système de redistribution

Durant cette séance, mon rôle était de câbler et testé notre algorithme de redistribution des pièces et d'apprendre à bien utiliser les Servo moteurs. Tout cela dans le but d'être prêt à monter une maquette pour la séance 4. La partie code a été faite par Alexis durant les vacances.

Voici le code :

```
1  #include <Servo.h>
2  int angleInitial=40;
3  int angleFinal=140;
4  Servo servo10;
5  Servo servo20;
6  Servo servo50;
7  Servo servo100;
8  Servo servo200;
9  int servo10Pos=angleInitial;
10 int servo20Pos=angleInitial;
11 int servo50Pos=angleInitial;
12 int servo100Pos=angleInitial;
13 int servo200Pos=angleInitial;
14
15 int val[]={200,100,50,20,10};
16 int nb[]={100,100,100,100,100}; //nombre de pieces presentes dans la machine
17 int aRendre[]={0,0,0,0,0}; //nombre de pieces à rendre
18 int argent=580; //argent que demande le client
19
20
21
22 void setup() {
23     Serial.begin(9600); //sert pour les tests
24     servo10.attach(2);
25     servo20.attach(3);
26     servo50.attach(4);
27     servo100.attach(5);
28     servo200.attach(6);
29     servo10.write(servo10Pos);
30     servo20.write(servo20Pos);
31     servo50.write(servo50Pos);
32     servo100.write(servo100Pos);
33     servo200.write(servo200Pos);
34     delay(1000);
35 }
36
37
38
39 int tot(int list[]){ //fonction qui permet de calculer l'argent total a partir d'une liste
40     int nb=0;
41     for (int i=0;i<5;i++){nb+=list[i]*val[i];}
42     return nb;
43 }
44
```

```

45 void piecesARendre(int euro){
46     for(int i=0;i<5;i++){
47         while (nb[i]>0 && tot(aRendre)+val[i]<=euro){
48             aRendre[i]+=1;
49             nb[i]--;}
50     }
51
52 int distrib(Servo servo,int pos, int nbPiece){
53     for (int j=0; j<nbPiece; j++){
54
55         if (pos==angleInitial){
56             for (int i = angleInitial; i <= angleFinal; i++) {
57                 servo.write(i);
58                 delay(5);}
59             pos=angleFinal;}
60         else{
61
62             for (int i = angleFinal; i >= angleInitial; i--) {
63                 servo.write(i);
64                 delay(5);}
65             pos=angleInitial;}
66         delay(500);}
67
68     return pos;
69 }
70
71
72
73
74 void loop() {
75     if (argent!=0){
76         piecesARendre(argent);          // ici mettre l'argent que demande le client
77
78     //serie de print pour les tests :
79     Serial.print(aRendre[0]);
80     Serial.print(" ");
81     Serial.print(aRendre[1]);
82     Serial.print(" ");
83     Serial.print(aRendre[2]);
84     Serial.print(" ");
85     Serial.print(aRendre[3]);
86     Serial.print(" ");
87     Serial.print(aRendre[4]);
88     Serial.print(" ");
89     Serial.println(tot(aRendre));
90     //

```

```

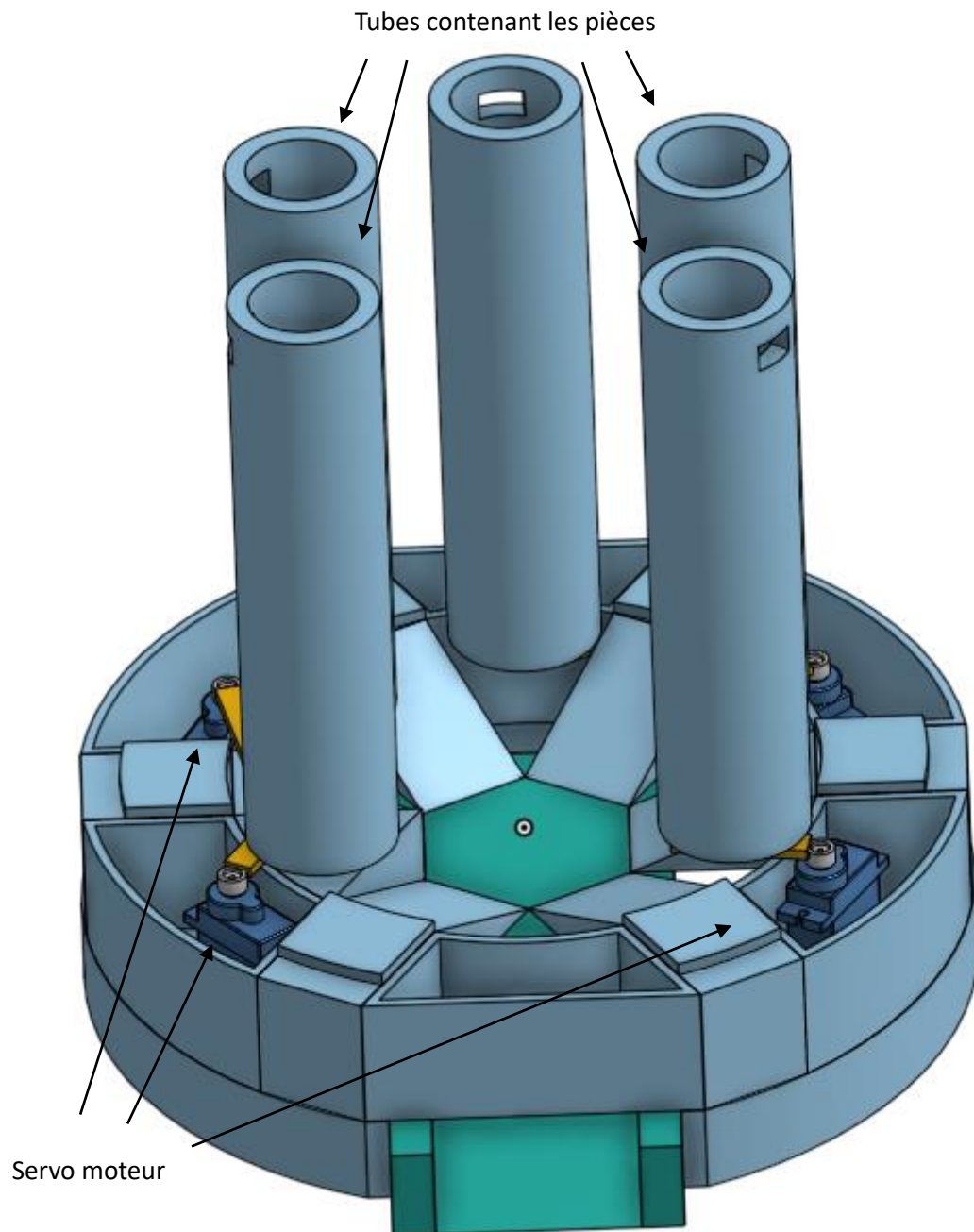
91
92     //on distribue les pieces
93     servo10Pos=distrib(servo10,servo10Pos,aRendre[4]);
94     servo20Pos=distrib(servo20,servo20Pos,aRendre[3]);
95     servo50Pos=distrib(servo50,servo50Pos,aRendre[2]);
96     servo100Pos=distrib(servo100,servo100Pos,aRendre[1]);
97     servo200Pos=distrib(servo200,servo200Pos,aRendre[0]);
98     argent=0;
99     for(int i=0;i<5;i++){aRendre[i]=0;}
100 }
101
102     delay(5000);
103 }

```

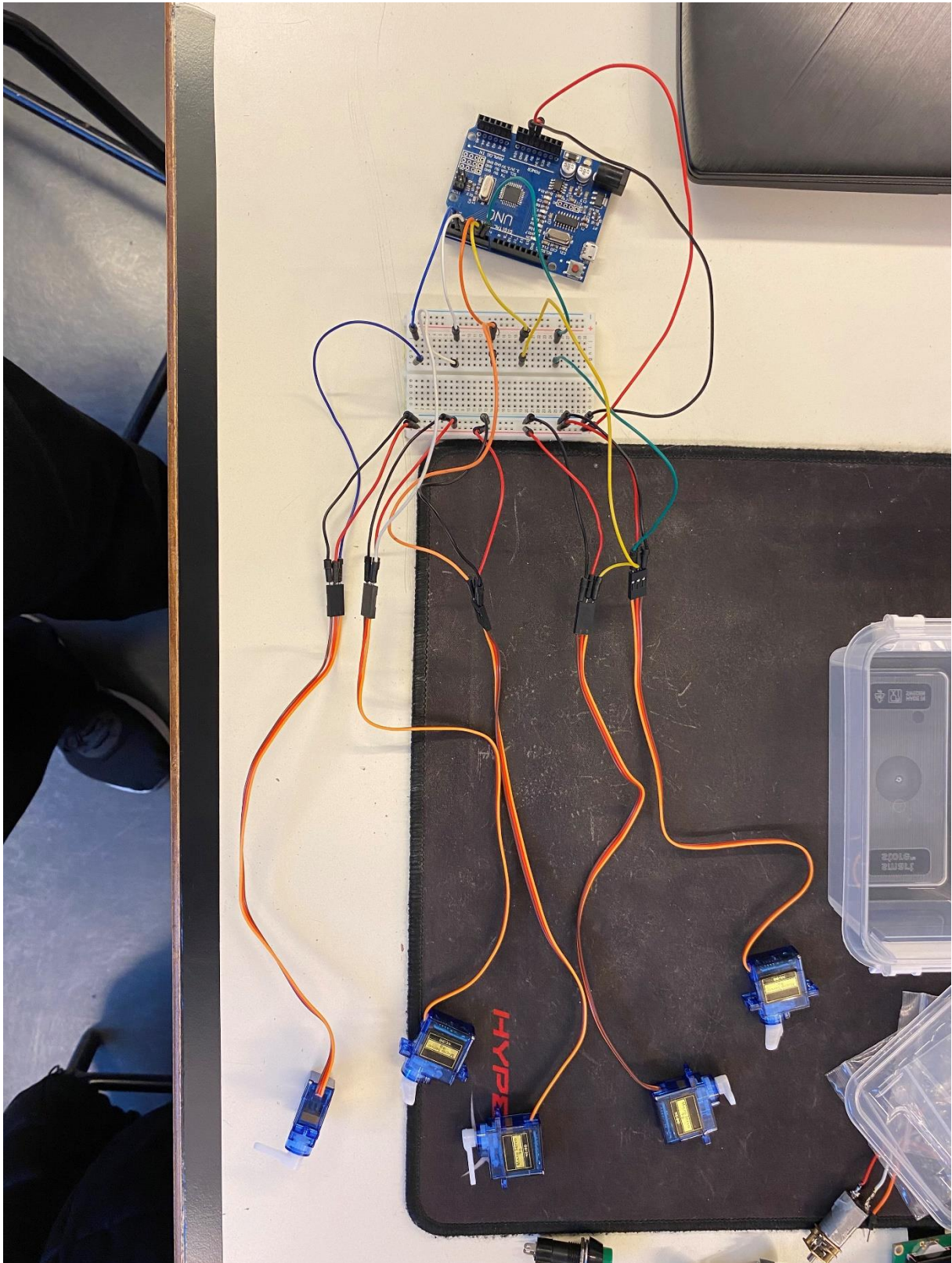
Il s'agit d'un algorithme qui se base sur 3 listes de valeurs et de la somme à rendre, une liste pour connaître le nombre de pièces présente dans la machine, une liste pour indiquer les valeurs des pièces et une liste indiquant le nombre de pièces à rendre pour chaque type de pièces.

L'algorithme va décomposer la somme à rendre en fonction des différentes pièces s'il voit que la machine en a en stock, il incrémente la valeur correspondante dans la liste des pièces rendues.

Ainsi, après coup, le programme va actionner les Servo moteurs correspondant aux pièces qui doivent être données. Ainsi les pièces tombent dans le bacs de rendu ?



Et donc voici le montage :



En fonctionnement : <https://youtube.com/shorts/racFUhVqIU?feature=shared>

Alexis ayant fait de son côté la communication Bluetooth, il reste l'implémentation des boutons et de l'écran LCD et de tout assembler sur une maquette avant de produire la version finale. L'assemblage du code ne semble pour l'instant pas si complexe mais un problème risque de surgir, nous risquons de ne pas avoir assez de ports d'entrées-sorties sur notre carte Arduino Uno.