AUSSEDAT Arthur G4

Rapport de séance 2

Pour cette séance j'ai décidé de commencer, voire finir, le code de la redistribution des pièces ainsi que celui des Servomoteurs si j'arriver à comprendre leur fonctionnement dans la séance.

Ce code de redistribution devra être capable de redonner les pièces en fonction de la somme demandée, sans choisir quelles pièces en particulier on souhaite ce qui serait bien plus facile à coder.

Pour le code, voici les variables initialisées.

```
int e200 = 0;
int e100 = 0;
int e50 = 0;
int e20 = 0;
int e10 = 0;

bool n200 = true;
bool n100 = true;
bool n50 = true;
bool n20 = true;
bool n10 = true;

int d = 0;
```

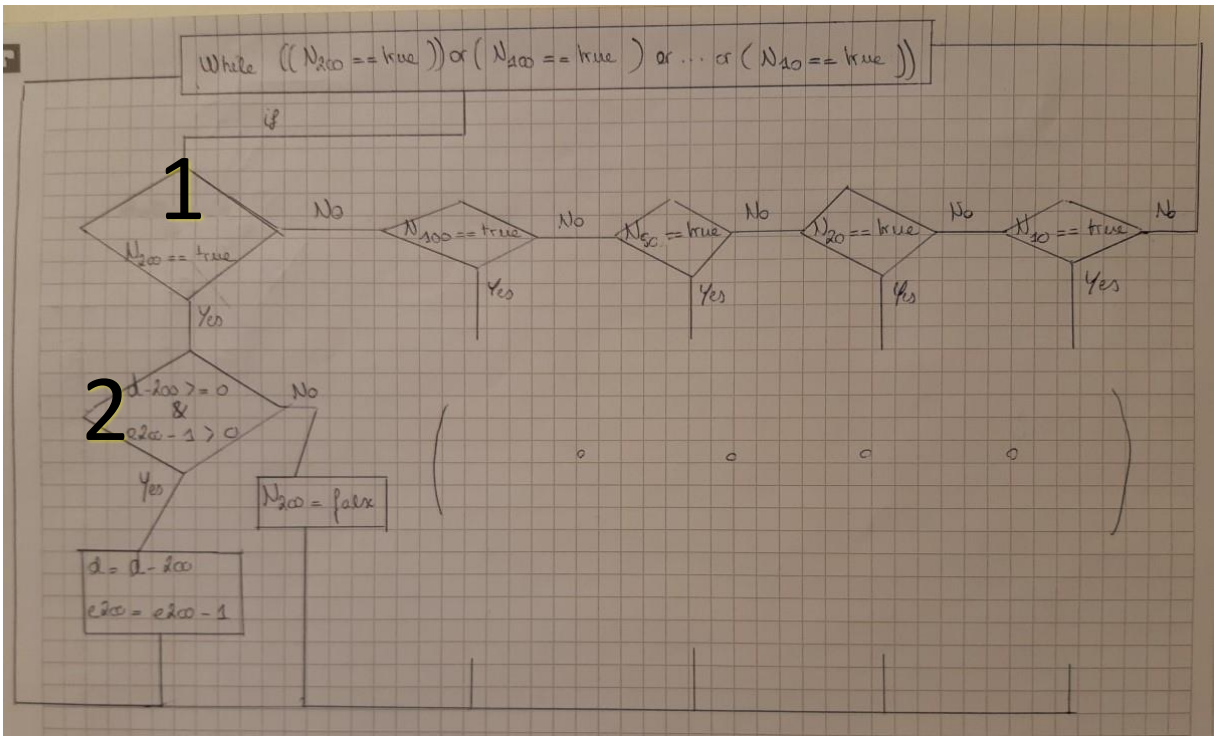
La variable « d » correspond à la somme demandée par l'utilisateur.

Les « int eXXX » contiennent le nombre de pièces présentes dans la tirelire en centime (e200 correspond au nombre de pièces de 2euros dans la tirelire, e50 → 50centimes, …)

Les « bool nXXX » correspondent au fait que le programme a bien vérifié que ce type de pièce pouvait être rendu ou non. Par exemple, si l'utilisateur demande 160 centimes, le programme va dans un premier temps vérifier s'il peut donner une pièce de 2euros, après avoir vérifié que non, il changera le booléen « n200 » en « false » ce qui signifie qu'il a vérifié ce type de pièce et peut passer à la suivante, le code ne s'arrête que lorsqu'il a vérifié chacune de pièces possibles.

Les variables « eXXX » seront modifiée par le capteur de présence qui augmentera ce nombre lorsqu'on ajoute une pièce et le code de redistribution ci-dessous fera l'inverse.

J'ai donc ensuite fait l'algorithme :



Je n'ai pas représenté la partie entre parenthèses car c'est presque la même chose que pour le « if n200 == true ».

Les variables e200, d et n200 sont expliquées juste au-dessus.

La partie 1 sert à vérifier que la pièce ne peut pas/plus être donnée avant de passer à la suivante car je pars du principe que la tirelire cherche toujours à rendre les pièces les plus grosses en premier.

La partie 2 sert à vérifier si : -la valeur de la pièce est bien inferieure ou égale à la somme demandée

-il reste bien au moins une pièce dans la tirelire

Si c'est le cas, on retire à la somme demandée, la valeur de la pièce et on retire une pièce au compteur, puis on retourne dans la boucle du While pour revérifier si on peut donner cette même pièce une seconde fois, ect.. Jusqu'à ce qu'on ne puisse plus.

J'ajouterai ensuite une 3eme action qui sera de rendre la pièce physiquement avec les servomoteurs.

Si ce n'est pas le cas, on sait que la pièce ne peut pas être donnée donc on change son nXXX en false pour le plus que le code passe dans sa boucle et on passe à la pièce suivante dans la prochaine boucle du While.

Voici donc le code :

```
void loop() {
  while((n200==true) or (n100==true) or (n50==true) or (n20==true) or (n10==true)){

    if (n200==true){

      if (d - 200 >= 0) & (e200 - 1 > 0){
        d -= 200;
        e200--;}
      else{n200 = false;}
    }

    else{

      if (n100==true){

        if (d - 100 >= 0) & (e100 - 1 > 0){
          d -= 100;
          e100--;}
        else{n100 = false;}
        }

      else{

        if (n50==true){

          if (d - 50 >= 0) & (e50 - 1 > 0){
            d -= 50;
            e50--;}
          else{n50 = false;}
        }

        else{
           if (n20==true){

             if (d - 20 >= 0) & (e20 - 1 > 0){
               d -= 20;
               e20--;}
             else{n20 = false;}
             }

           else{

              if (n10==true){

                if (d-10 >= 0) & (e10 - > 0){
                 d -= 10;
                 e10--;}
                else{n10==false;}
              }
           }
        }
      }
    }
  }
}
```

Evidemment il manque la partie qui active le servomoteur qui correspond à chaque pièce, cette dernière sera faite lorsque j'aurai fait une maquette.

Il manque maintenant une dernière partie à ce code : que faire lorsque la somme demandée ne peut pas être exactement rendue. Il faut que la machine rende toujours plus que la somme demandée dans ce cas.

J'ai donc fait ce code à la suite du précédent :

```
int m = 0; //manque: argent que la machine n'a pas pu rendre
int r = 0; //rajout: argent que la machine à donné en trop
```

Premièrement j'ai ajouté deux nouvelles variables

Et voici le code :

```
if (d != 0){ //la machine n'a pas pu rendre le montant exact
  while(d - r > 0){//le rajout doit couvrir ce qu'il manque
    if (e10 > 0){
      e10--;
      r = d + 10;}

      else{
        if (e20 > 0){
          e20--;
          r = d +20;}

        else{
          if (e50 > 0){
            e50--;
            r = d +50;}

          else{
            if (e100 > 0){
              e100--;
              r = d+100;}

            else{
              if (e200 > 0){
                e200--;
                r = d+200;}

              else{
                m = d;
              }
```

On ne rentre dans cet if que si : d est différent de 0, donc la somme demandée n'a pas pu être rendue exactement.

Le while sert à s'assurer qu'on rend plus que la somme demandée. Puis on rend des pièces des plus petites jusqu'aux plus grandes jusqu'à qu'on ait rendu plus que ce qui était demandé.

J'ai quand même ajouté une variable m qui ne sera utilisée que s'il est impossible de répondre à la demande de l'utilisateur. Cette dernière servira sur l'écran LCD.

Ce code m'a demandé la majorité de la séance à imaginer et réaliser, j'ai donc, pendant le temps restant, commencé à me familiariser avec le fonctionnement des servomoteurs et leur code. Je rencontre toujours quelques difficultés à les comprendre je vais donc m'en servir pendant les vacances et essayer d'écrire leur code pour rendre les pièces même.