Ugo Morena G2

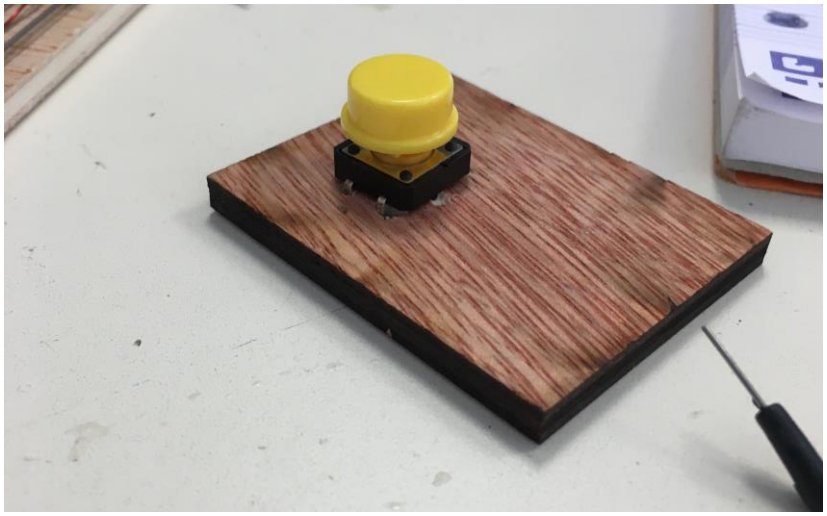
RAPPORT SÉANCE 10 FEVRIER 2020

Objectif de la séance :

- **Création de la trappe avec le moteur + ajout au code**
- **Test de la conceptualisation des boutons du mastermind sur un bouton pour voir si la solution est viable**
- **Finalisation soudure LEDS**
- **Test Mastermind au complet !**

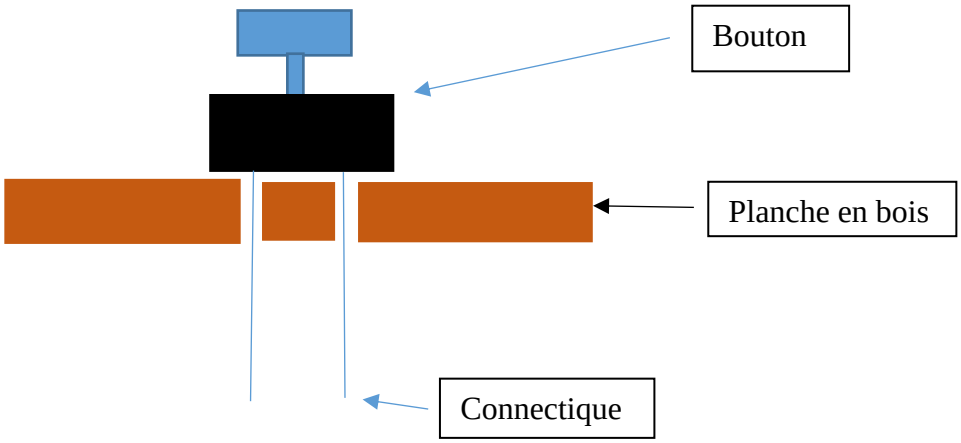
Test sur le premier bouton :

J'ai pris une simple planche en bois un peu épaisse pour pouvoir poser le bouton dessus et percer la planche afin de faire passer les pins du bouton. J'ai ensuite soudé le dessous du bouton pour pouvoir traverser la planche.



Voici le rendu.

Ensuite, j'ai fait quelques dessins de mon côté afin d'expliquer comment vont se disposer les boutons, voici un récap :



Ugo Morena G2

RAPPORT SÉANCE 10 FEVRIER 2020

Par la suite, nous rajouterons un élément esthétique (une fine planche) entre le bouton et son support (respectivement bleu et noir sur la photo).

Création de la trappe :

J'ai découpé puis percé dans un premier temps une planche en bois, à laquelle j'ai fixé le moteur en perçant sur les côtés de la planche. Voici un petit rendu :

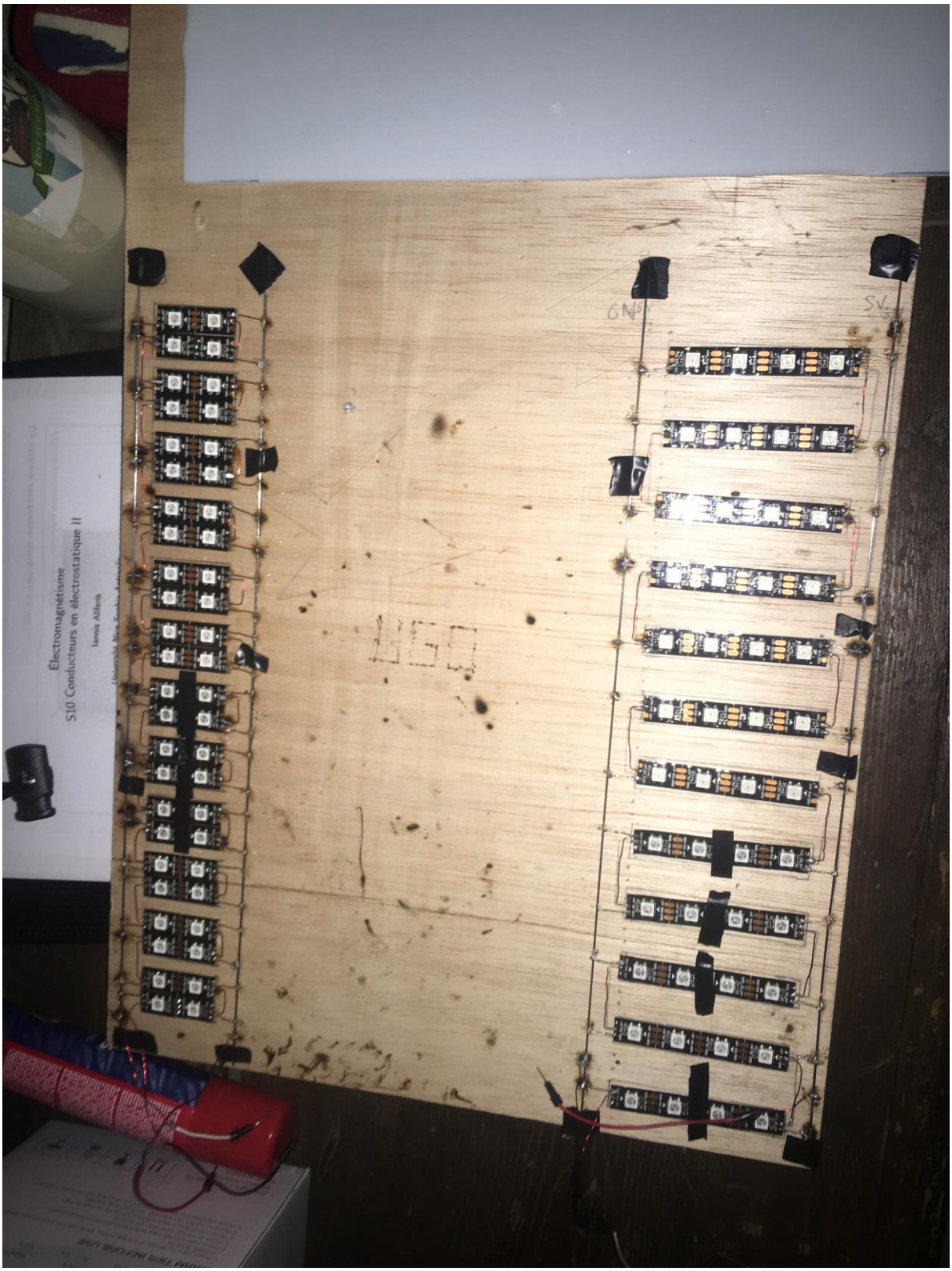
J'ai ajouté au code un mouvement du moteur quand on trouve la solution, où quand la partie est finie. La trappe se referme toute seule au lancement d'une nouvelle partie si celle-ci est ouverte !

Finalisation soudure LEDS :

J'ai ensuite pris du temps pour pouvoir souder toutes les LEDS restantes, et j'ai enfin pu terminer après avoir dû ressouder environ 24 LEDS pour cause de mauvaises soudures. Voici le rendu final des LEDS, il ne reste donc plus que la boite avec le socle.

Ugo Morena G2

RAPPORT SÉANCE 10 FEVRIER 2020



Pour voir le rendu final, je vous joins sur le GitHub une petite vidéo artisanale d'une partie.

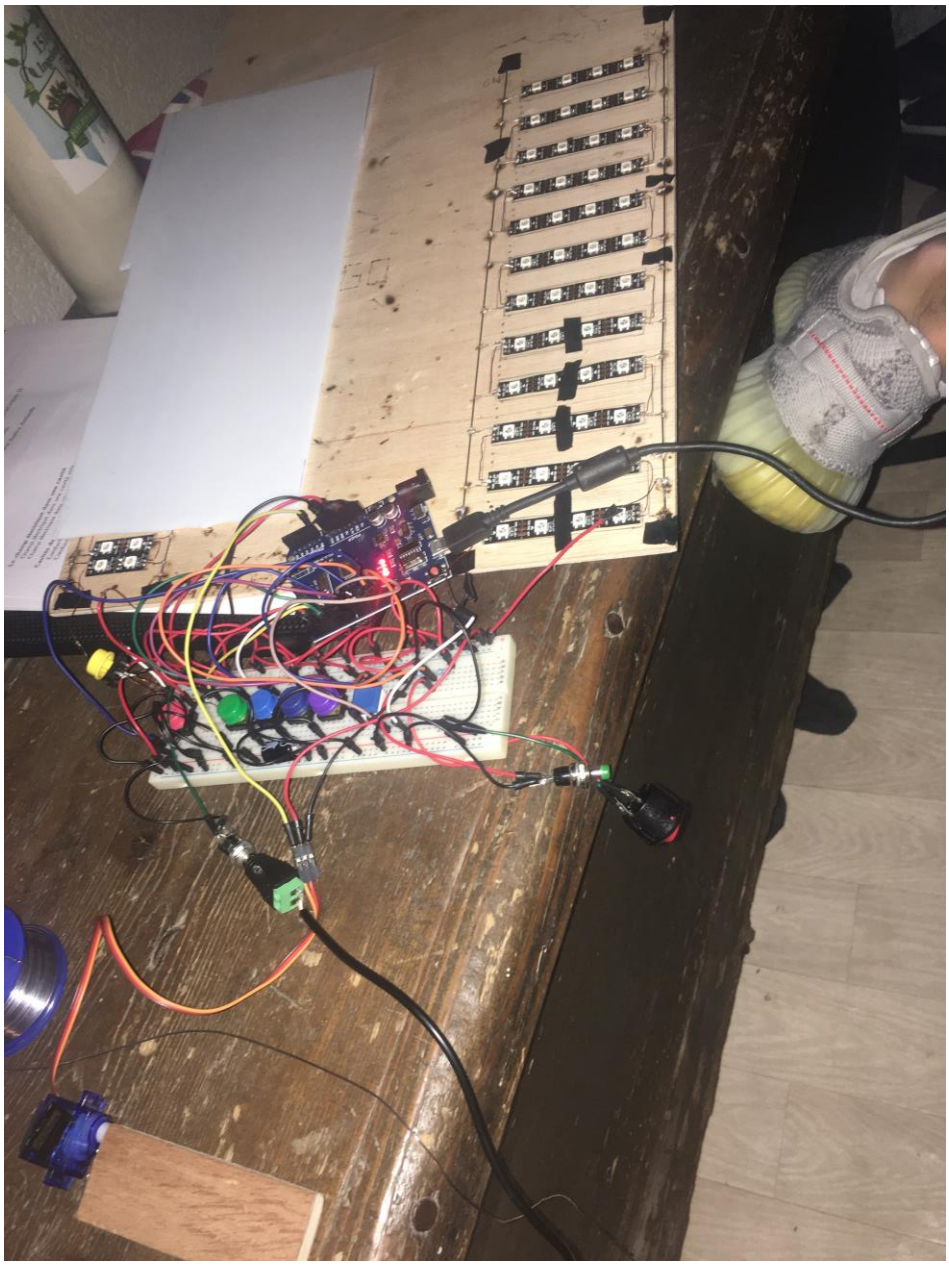
Ugo Morena G2

RAPPORT SÉANCE 10 FEVRIER 2020

J'ai passé tout le reste de la séance à préparer la création de la boite du Mastermind et améliorer le code. Nous allons au FabLab le Vendredi 14/02 pour commencer la fabrication.

Je vous joins le code tout en bas au cas où vous voudriez jeter un coup d'œil (il est plutôt long).

A la prochaine séance il nous restera la finalisation de l'application et la jonction avec le Mastermind, mais aussi l'ajout d'un module audio pour jouer des musiques.



Ugo Morena G2

RAPPORT SÉANCE 10 FEVRIER 2020

CODE :

```
#include <Adafruit_NeoPixel.h>

#include <Servo.h>

Servo trappe;
```

RAPPORT SÉANCE 10 FEVRIER 2020

```
const int b_rouge=2; //pin boutons

const int b_jaune=3;

const int b_vert=4;

const int b_cyan=5;

const int b_bleu=6;

const int b_violet=7;


boolean finDuJeu = false;


const int restart=11;

const int gm=12; //gamemode

const int undo=13;


String etat;

String etatTab[] = {"solo","duo"};


int i=0; //ligne

int j=0; //colonne


int nbBouton=6;

int boutonListe[6]= {2,3,4,5,6,7};

int b_color[6][3]={{255,0,0},{255,255,0},{0,255,0},{0,255,255},{0,0,255},{155,0,255}};
//permet de connaitre la couleur en fonction du bouton (classé dans le même ordre que
les boutons)



int
ledAnaly[12][4]={{0,1,3,2},{4,5,7,6},{8,9,11,10},{12,13,15,14},{16,17,19,18},{20,21,23,22},{
```

```
24,25,27,26},{28,29,31,30},{32,33,35,34},{36,37,39,38},{40,41,43,42},{44,45,47,46}};//liste de toutes les leds

int led[12][4]={{0,1,2,3},{7,6,5,4},{8,9,10,11},{15,14,13,12},{16,17,18,19},{23,22,21,20},{24,25,26,27},{31,30,29,28},{32,33,34,35},{39,38,37,36},{40,41,42,43},{47,46,45,44}};

int ledSol[4]={0,1,2,3}; //leds solution (dans la trappe)


const int LED_PIN=8;//pin leds

const int LED_PIN_SOL=9;//pin leds sol

const int LED_ANALY_PIN=10; //pin leds analyseuses


const int NUM_LEDS=48; //nb leds

const int NUM_LEDS_SOL=4;//nb leds sol

const int NUM_LEDS_ANALY=48; //nb leds analyseuses


int vert=0; //nb de LEDS allumées pour bonne couleur bon placement

int rouge=0; //nb de LEDS allumées pour bonne couleur mauvais placement


int reponse[12][4]; //liste de toutes les réponses (ici on garde pour chaque ligne)

int sol[4]={9,9,9,9}; //etat initial, 9 ne correspond à aucune des valeurs possibles

int solVar[4]; //liste temporaire des solutions (voir suite du code)

int reponseVar[4]; //liste temporaire des réponses (voir suite du code)



Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUM_LEDS,LED_PIN,NEO_GRB + NEO_KHZ800); //bande de leds

Adafruit_NeoPixel stripSol = Adafruit_NeoPixel(NUM_LEDS_SOL,LED_PIN_SOL,NEO_GRB + NEO_KHZ800); //bande de leds solution (trappe)
```

RAPPORT SÉANCE 10 FEVRIER 2020

```
Adafruit_NeoPixel stripAnaly = Adafruit_NeoPixel(NUM_LEDS_ANALY,LED_ANALY_PIN,NEO_GRB + NEO_KHZ800); //bande de leds analyseuses



void setup() {


  Serial.begin(9600);

  trappe.attach(14);

  i=0;

  j=0;

  trappe.write(190);

  pinMode(b_rouge,INPUT);

  pinMode(b_vert,INPUT);

  pinMode(b_jaune,INPUT);

  pinMode(b_cyan,INPUT);

  pinMode(b_bleu,INPUT);

  pinMode(b_violet,INPUT);

  pinMode(restart,INPUT);

  pinMode(gm,INPUT);

  stripSol.begin();

  stripAnaly.begin();

  strip.begin();

  stripSol.setBrightness(70);

  strip.setBrightness(70);

  strip.show();

  stripAnaly.setBrightness(70);

  stripAnaly.show();
```

```
 stripSol.show();
 int l=0;
 int cpt=0;
 for (int k=0; cpt<NUM_LEDS;k++){ //pour allumer toutes les leds au démarrage
  if (k==4){
    l+=1;
    k=0;
  }
  strip.setPixelColor(led[l][k],100,100,100);
  cpt+=1;
}
cpt=0;
l=0;
 for (int k=0; cpt<NUM_LEDS;k++){ //pour allumer toutes les leds au démarrage
  if (k==4){
    l+=1;
    k=0;
  }
  stripAnaly.setPixelColor(ledAnaly[l][k],100,100,100);
  cpt+=1;
}
cpt=0;
 for (cpt;cpt<4;cpt++){
  stripSol.setPixelColor(ledSol[cpt],100,100,100);
}
 strip.show();
 stripSol.show();
```

```
  stripAnaly.show();


  delay(1000);


  strip.clear();
  stripAnaly.clear();
  stripSol.clear();
  strip.show();
  stripSol.show();
  stripAnaly.show();
  //delay(1000);
  etat=etatTab[digitalRead(gm)];


}


void loop() {
  //Serial.println(i);
  //Serial.println(digitalRead(gm));
  //Serial.println(etat);



 /* RESTART */
  if (digitalRead(restart)==LOW) {


    for (int i=0;i<4;i++){
      sol[i]=9;
    }
```

```
  strip.clear(); //efface tout
  stripSol.clear();
  stripAnaly.clear();
  strip.show();
  stripSol.show();
  stripAnaly.show();
  i=0; //valeur à 0
  j=0;
  while (digitalRead(restart)==LOW){} // tant qu'on appuye dessus
  Serial.println("restart");
  delay(500);
  }

 /* CHANGEMENT DE JOUEUR */
 if (etat!=etatTab[digitalRead(gm)]){ //si l'etat stockant le nb de joueurs à changer en cours de jeu
  Serial.println("gm change");
  for (int gmVar=0;gmVar<4;gmVar++){
    sol[gmVar]=9;
  }

  stripSol.clear(); //on remet tous à zéro
  strip.clear();
  stripAnaly.clear();
  stripSol.show();
  strip.show();
  stripAnaly.show();
```

```
  i=0;
  j=0;
  etat=etatTab[digitalRead(gm)]; // on réactualise le nb de joueur
 }

 /* GENERATION SOLUTION SOLO/DUO */

 if (sol[0]==9){ //initialement, forcément à 9
  if (etat=="solo"){ //si joueur en solo
  Serial.println(etat);
  for (int b=0;b<4;b++){
    int r=random(0,5); //génération d'un chiffre aléatoire
    stripSol.setPixelColor(ledSol[b],b_color[r][0],b_color[r][1],b_color[r][2]); //puis on allume la led numero b, de la couleur trouvé en c (numéro du bouton dans la liste)
    stripSol.show();
    sol[b]=boutonListe[r]; //ajout du bouton, correspondant à l'emplacement du chiffre, à la liste solution
    Serial.println(sol[b]);
    }
  } //fin if

  else if (etat=="duo"){
    //Serial.println(etat);
    delay(500);
    int b=0; //variable permettant de compter les ledSol
    while (sol[b]==9){
      for (int a=0;a<nbBouton;a++){ //on parcourt la liste des boutons
        if (digitalRead(boutonListe[a])==LOW){ //dès qu'une valeur est à high
```

```
          //Serial.println(boutonListe[a]); //on l'affiche pour vérifier que tout marche
          for (int c=0;c<nbBouton;c++){ //boucle permettant de parcourir les couleurs des boutons
            if (boutonListe[a]==boutonListe[c]){ //on repère quel bouton est cliqué
              //Serial.println(boutonListe[c]); //affichage pour vérifier
              stripSol.setPixelColor(ledSol[b],b_color[c][0],b_color[c][1],b_color[c][2]); //puis on allume la led numero b, de la couleur trouvé en c (numéro du bouton dans la liste)
              stripSol.show();
              sol[b]=boutonListe[a]; //on ajoute le bouton cliqué à la liste solution
              Serial.println(sol[b]);
            }
          }
          b+=1;
          while (digitalRead(boutonListe[a])==LOW){} //tant qu'on appuye sur ce bouton

        } //fin if le bouton est appuyé
      } //fin for
    } // fin while
  } //fin else if
} //fin if sol[0]==9

/* JEU PRINCIPAL */
else { // Si la solution existe déjà, c-a-d que la partie est en cours
  //Serial.println("début");
  if (digitalRead(undo)==LOW){ //pour annuler le mouvement précédent
    if (j!=0){ //si on est pas au début de ligne
      j-=1; //on revient à la dernière colonne
```

```
      strip.setPixelColor(led[i][j],0,0,0); //et on éteint la led jouée
      strip.show(); //on l'affiche
      while (digitalRead(undo)==LOW){} //tant qu'on appuye rien ne se passe
      }
   }
   for (int a=0;a<nbBouton;a++){ //on parcourt la liste des boutons
    if (digitalRead(boutonListe[a])==LOW){ //dès qu'une valeur est à high
      //Serial.println(boutonListe[a]); //on l'affiche pour vérifier que tout marche
      for (int c=0;c<nbBouton;c++){
       if (boutonListe[a]==boutonListe[c]){
         //Serial.println(boutonListe[c]);
         strip.setPixelColor(led[i][j],b_color[c][0],b_color[c][1],b_color[c][2]);
         strip.show();
        }
      }
      while (digitalRead(boutonListe[a])==LOW){} //tant qu'on appuye sur le bouton rien ne se passe et le programme est en "pause"
      reponse[i][j]=boutonListe[a]; //dès qu'on relache le bouton, le programme reprend et on ajoute la réponse à la liste réponse.
      Serial.println(reponse[i][j]);
      j+=1;
     }
   }

   /* FIN TOUR */
   if (j==4){ //si on est à la fin de la ligne
    for (int sVar=0;sVar<4;sVar++){ //on stocke les valeurs dans une liste temporaire, pour pouvoir supprimer la valeur une fois qu'elle a été analysée
```

```
    solVar[sVar]=sol[sVar];
  }
  for (int rVar=0;rVar<4;rVar++){ //idem pour la réponse
    reponseVar[rVar]=reponse[i][rVar];
  }

  /* COMPARAISON REPONSE SOLUTION */
  vert=0;
  rouge=0;
  for(int a=0;a<4;a++){ //chaque élément des réponses va parcourir la liste de la solution et comparer les valeurs
    for (int b=0;b<4;b++){
      if ((reponseVar[a]==solVar[b]) && (a==b)){ //si bonne réponse bon emplacement
        Serial.println("vert");
        vert+=1;
        solVar[b]=9; //on supprime de la liste les valeurs qui ont été analysées pour ne pas fausser le résultat, cad ne pas compter deux fois la même valeur
        reponseVar[a]=10; //comme par exemple si la solution est (1,1,1,1) et qu'on rentre (1,2,3,4), il faut compter le 1 qu'une seule fois
      }
    }
  }
  for(int a=0;a<4;a++){ //maintenant on refait mais pour connaitre les bonnes couleurs mauvais emplacements
    for (int b=0;b<4;b++){
      if ((reponseVar[a]==solVar[b]) && (a!=b)){ //bonne couleur mauvais emplacement
        Serial.println("rouge");
        rouge+=1;
```

```
      solVar[b]=9; //cette fois-ci on remplace uniquement celle de la solution par 9
    }
  }
}

/* AFFICHAGE DES RESULTATS */
int jA=0;
for (int vertVar=0;vertVar<vert;vertVar++){
  stripAnaly.setPixelColor(ledAnaly[i][jA],0,255,0);
  jA+=1;
}
for (int rougeVar=0;rougeVar<rouge;rougeVar++){
  stripAnaly.setPixelColor(ledAnaly[i][jA],255,0,0);
  jA+=1;
}
stripAnaly.show();

if (vert==4){
  finDuJeu=true;
  Serial.println("win");
  for (int position = 190; position >=80;position --){
    trappe.write(position);
    delay(10);
  }
  //********musique loose
}
else {
```

```
      i+=1; //on passe à la ligne suivante
      Serial.println(i);
      j=0; //on revient à la première colonne
    }

    if (i==12){
      finDuJeu=true;
      for (int position = 190; position >=80;position --){
        trappe.write(position);
        delay(10);
      }
      //********ouverture trappe
      //********musique loose
    }

    if (finDuJeu==true){
      while(digitalRead(restart)==HIGH){} //tant qu'on appuye pas sur le bouton restart rien ne se passe
      for (int i=0;i<4;i++){ //on recommence tout (on a appuyé sur le bouton pour être ici
        sol[i]=9;
      }

      trappe.write(190);
      finDuJeu=false;
      strip.clear(); //efface tout
      stripSol.clear();
      stripAnaly.clear();
```

```
      strip.show();
      stripSol.show();
      stripAnaly.show();
      i=0; //valeur à 0
      j=0;
      while (digitalRead(restart)==LOW){} // tant qu'on appuye dessus
      Serial.println("restart");
      delay(500);
    }//fin si finDuJeu
  }//fin si fin de ligne
 }//fin else (la solution existe déjà)
}//fin void loop
```