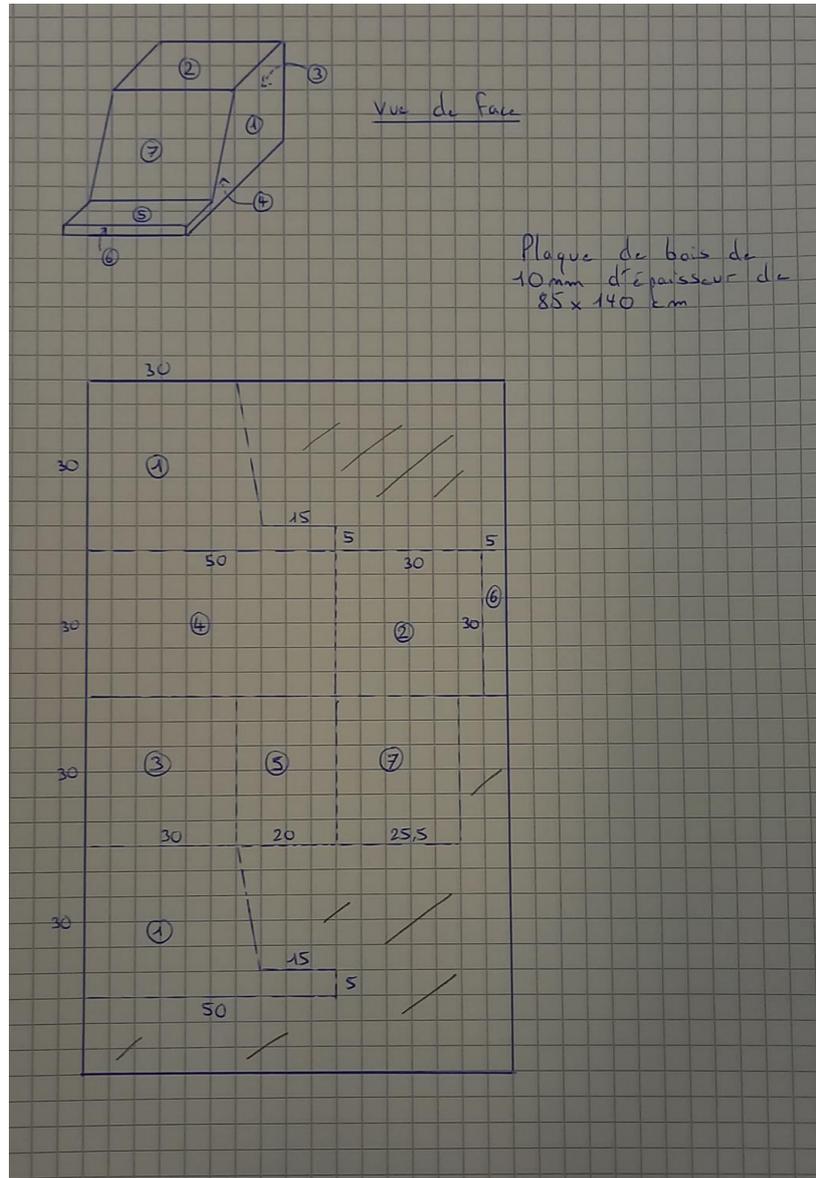


COMPTE RENDU DU POC

Le but de notre POC est de fabriquer un prototype visuel de notre installation et de programmer un système de remise de point selon une échelle qui sera étudiée.

1ère Étape : Réalisation du prototype

Pour ce faire, nous avons réaliser un plan de construction pour notre support qui sera constitué d'une balance et d'un écran intégré.



L'écran coûtant cher, nous afficheront les points, données sur un écran d'ordinateur à côté pour la présentation. On disposera une balance au dessus de la boîte, celle-ci devrait être intégré dans le l'idée du système final.

Le résultat du prototype sera présenté dans la partie Bilan du POC.

2ème Étape : Étude d'une échelle adaptée

Nous avons déterminé une échelle adaptée en fonction des gâchis dans les cantines, nous avons recherché sur internet le gâchis moyen d'un collégien et avons donc pu établir celle-ci.

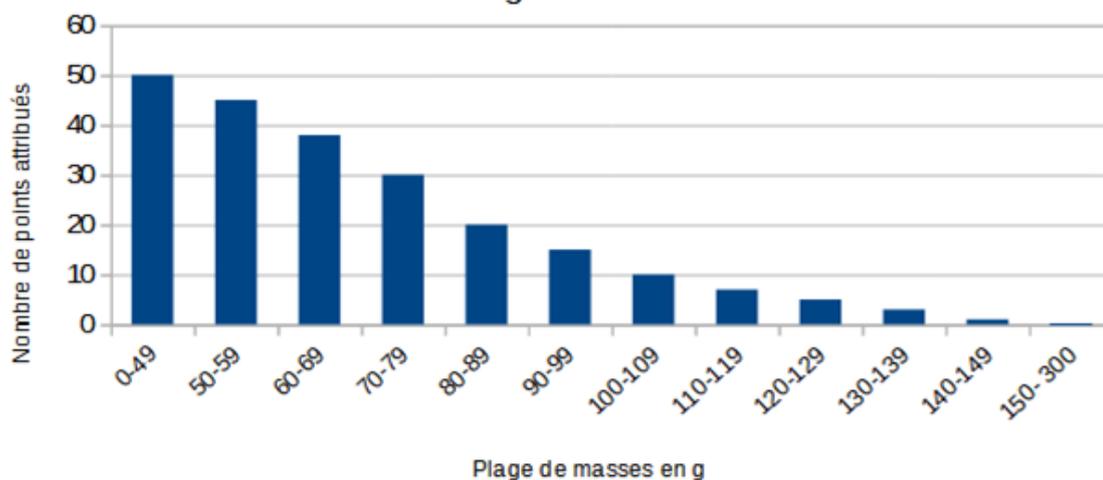
Les idées directrices sont :

- Nous établiront un nombre de points selon une plage de poids, exemple entre 0 et 50g l'élève gagnera 50 points. Ceci permet d'éviter des erreurs comme enlever des points à un élève à cause du poids des os.

-Nous ne voulons pas une évolution linéaire des points en fonction du poids, le but est de sensibiliser sur le zéro déchet, nous adapteront donc le système de point pour que même si l'élève gâche un peu, il perd beaucoup de point par rapport à un camarade ayant tout mangé.

Plages de Masses en g	points
0-49	50
50-59	45
60-69	38
70-79	30
80-89	20
90-99	15
100-109	10
110-119	7
120-129	5
130-139	3
140-149	1
150- 300	0

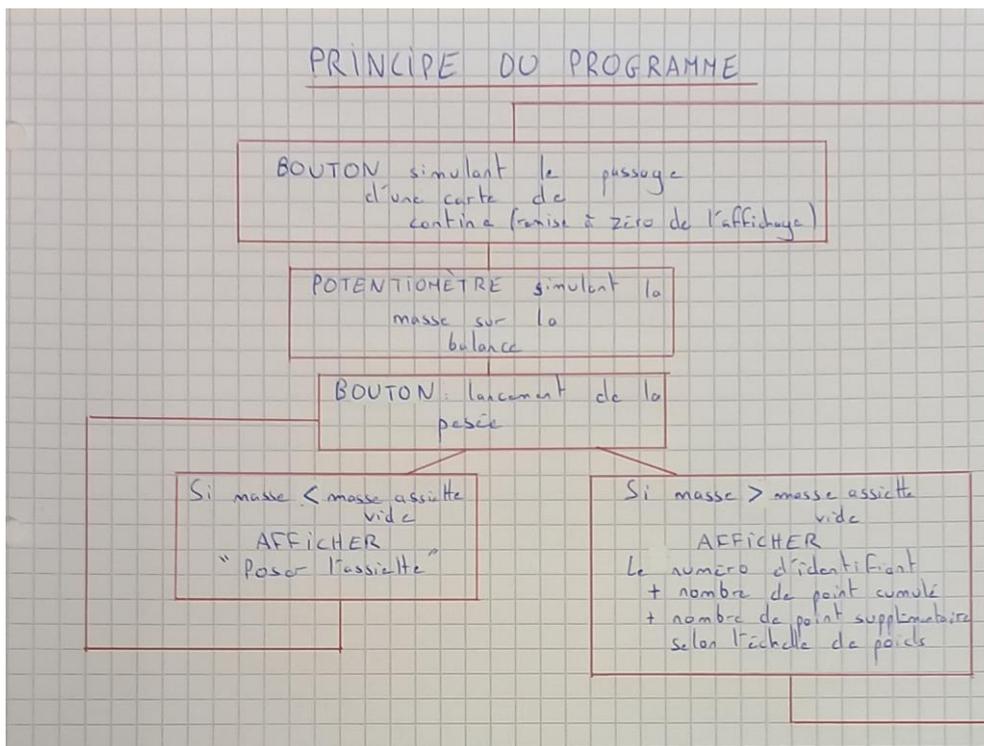
Diagramme des points attribuant en fonction de la masse de gâchis



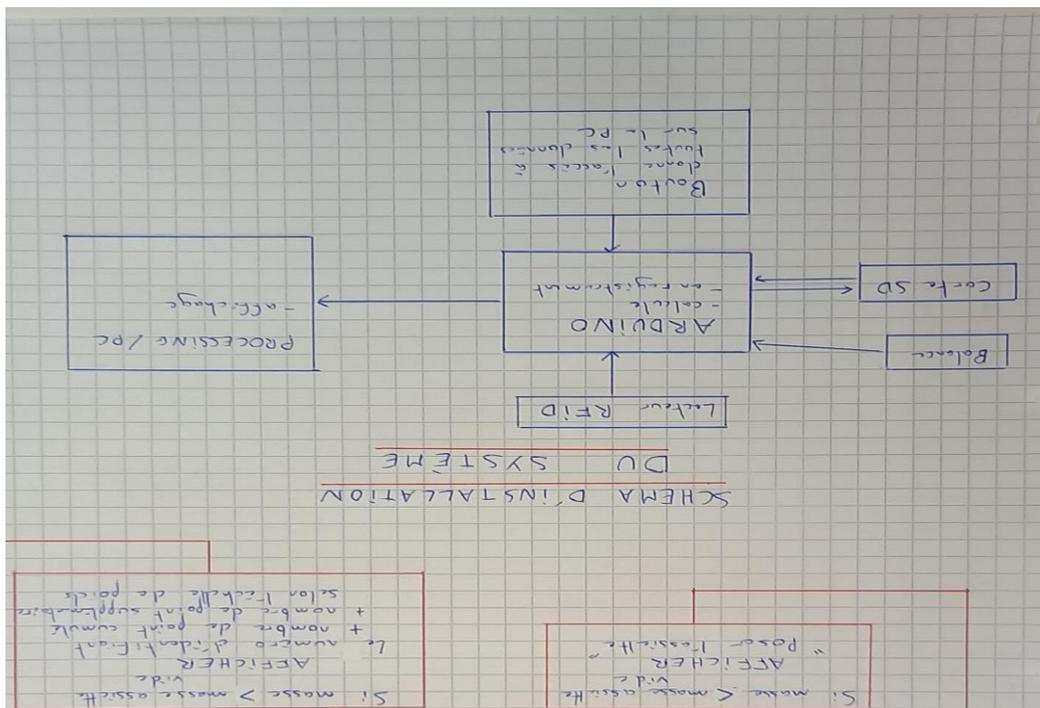
Cette échelle serait dans le principe mis dans le programme pour ainsi donner à l'élève les points qu'il mérite.

Étape 3 : Programmation et Affichage

Le principe de programmation est le suivant :



Pour effectuer les communications et ainsi programmer ceci, nous avons fait l'installation suivante :



Pour finir voici le code fait sur Arduino pour ainsi simuler notre concept et afficher les données nécessaires à son utilisation :

```

//Mesure masse
int Poids;

//identifiant
byte even = 0; //detection de badge
int ID = 0;

//Nombre de points
int Points = 700;

//Base de donnée : ID,PRENOM,NOM,POINTS
int DBpoids[3][2]={
  {38,0},
  {39,0},
  {40,0}
};
/*String DBname[3][2]={
  {"Fabien","Paquet"},
  {"Emmanuel","Echeverri"},
  {"Claire","Garcia"}
};*/

void setup() {
  Serial.begin(9600); // Initialise la communication avec l'ordinateur

  pinMode(38,INPUT); //bouton identifiant 1
  pinMode(39,INPUT); //bouton identifiant 2
  pinMode(40,INPUT); //bouton identifiant 3

  pinMode(A4, INPUT); //mesurer poids (potar)
}

void loop() {
  while(even == 0){
    if(digitalRead(38) == 1){
      even = 1;
      ID = 38;
      Serial.println("38");
    }
    else if(digitalRead(39) == 1){
      even = 1;
      ID = 39;
      Serial.println("39");
    }
    else if(digitalRead(40) == 1){
      even = 1;
      ID = 40;
      Serial.println("40");
    }
  }

  even = 0;

  Poids = analogRead(A4);

  while(Poids < 100){
    Serial.println("Posez l'assiette");
    delay(1000);
    Poids = analogRead(A4);
  }
}

```

```

DBpoids[ID-36][1] = DBpoids[ID-36][1] + Poids - 100;

/*Serial.print("Nom:");
Serial.print(DBname[ID-36][1]);
Serial.print("Prenom:");
Serial.print(DBname[ID-36][0]);*/
Serial.print(" Poids:");
Serial.print(Poids);
Serial.print(" Nombre de points:");
Serial.println(DBpoids[ID-36][1]);

delay(1000);
}

```

PROGRAMME PROCESSING

```

import processing.serial.*;
Serial myPort;
String stringLecture = "";

int poids;
int ID;

void setup() {

  size(1000, 400);

  // List all the available serial ports
  println(Serial.list());
  myPort = new Serial(this, Serial.list()[1], 9600); // à modifier en indiquant votre numéro de port série
  myPort.bufferUntil('\n'); // ne pas appeler serialEvent () sauf si un caractère de nouvelle ligne est présent dans le
  buffer
}

```

Grâce à ces trois étapes nous avons réussi à produire une preuve de concept qui comprends un prototype visuel avec une balance et un socle, on peut ainsi peser une masse de gâchis, l'écrire dans le programme et laisser celui-ci afficher les données utiles. Pour finaliser l'idée il faudrait réussir à transmettre directement l'information de masse de la balance dans le programme mais par manque de temps nous ne l'avons pas exploité.