

DIEVAL Cécile

Semaine 1 : projet porte-savon

Le but du projet est de réaliser un porte-savon en 3D.

Nous allons décomposer sa structure en deux parties :

- une grille en 2D via le logiciel **Inkscape**

Inkscape

- un contenant via le logiciel **Tinkercad**

Tinkercad

Nous allons imprimer ce porte-savon :

- via le logiciel **ideaMaker**

ideaMaker

- via le logiciel **Ultimaker Cura**

Ultimaker Cura

Pour ma part, j'ai choisi de réaliser un porte-savon version Nintendo Mario Bros - Course aux pièces.

1.Modélisation 2D de la grille

a. Régler la longueur et la largeur de la feuille

Action : aller dans l'onglet fichier puis dans propriété du document

Dimensions : 22 cm(longueur) X 18cm(largeur);et orientation=horizontal

b. Dessiner la forme de la grille

Action : prendre la forme rond

c. Dimensionner la forme

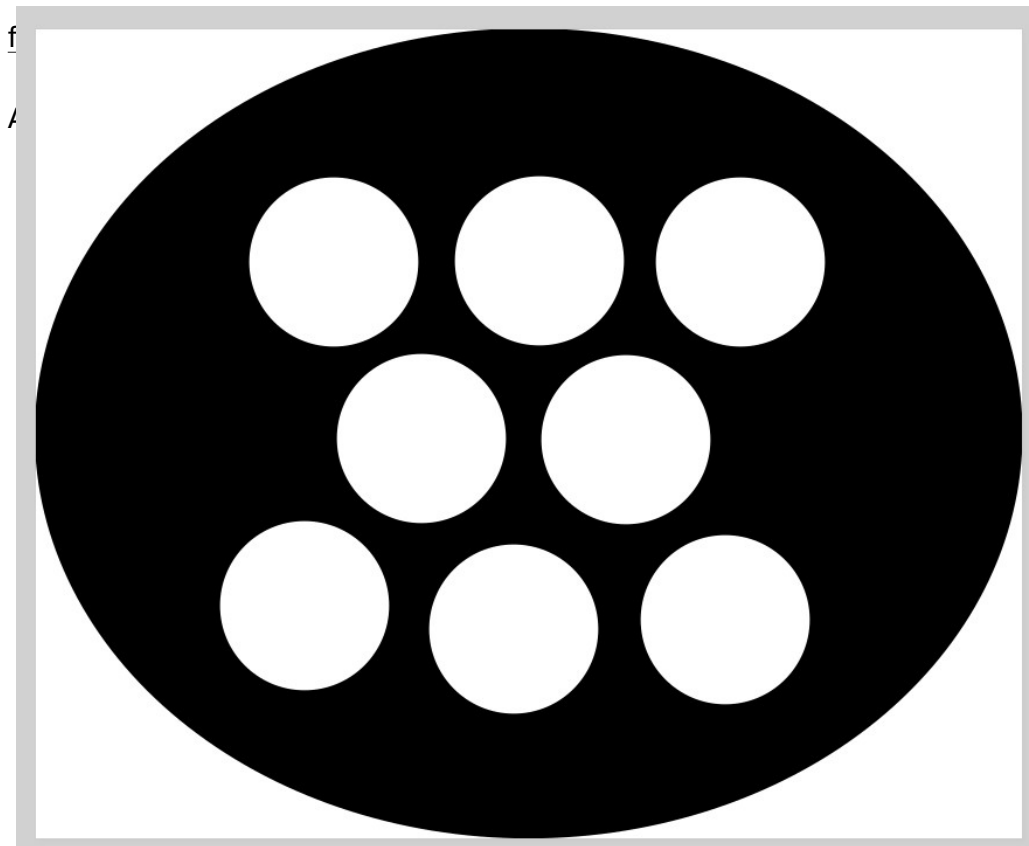
Action : $R_x = 9\text{cm}$ et $R_y = 7\text{cm}$

d. Mettre des petit ronds pour construire la grille

Action : prendre la forme rond

e. Dimensionner le rond

Action : $R_x = 2\text{cm}$ et $R_y = 2\text{cm}$



On aurait pu mettre un logo en plus, pour cela, il faut aller dans fichier puis importer une image; et également rajouter une épaisseur si besoin (cf.contour).

Fichier grille.svg ci-joint

2.Modélisation 3D du contenant

a. Importer le fichier sur Inkscape sur Tinkercad

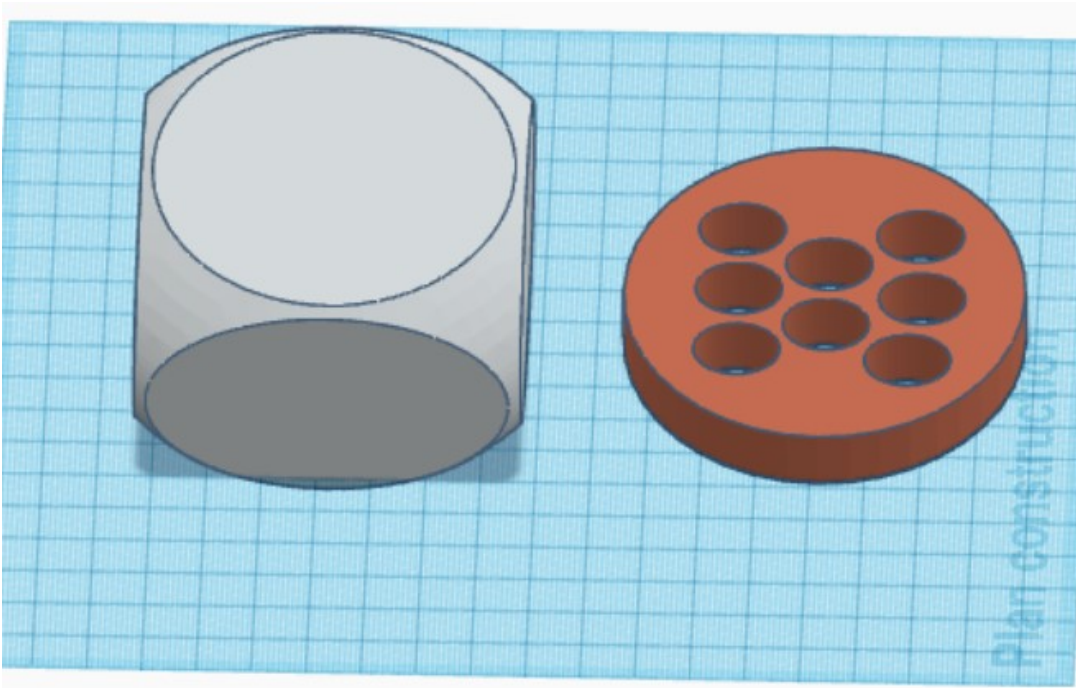
Action : importer le fichier .svg

Dimensions : 90 mm (longueur) x 70 mm (largeur); et 10mm (épaisseur)

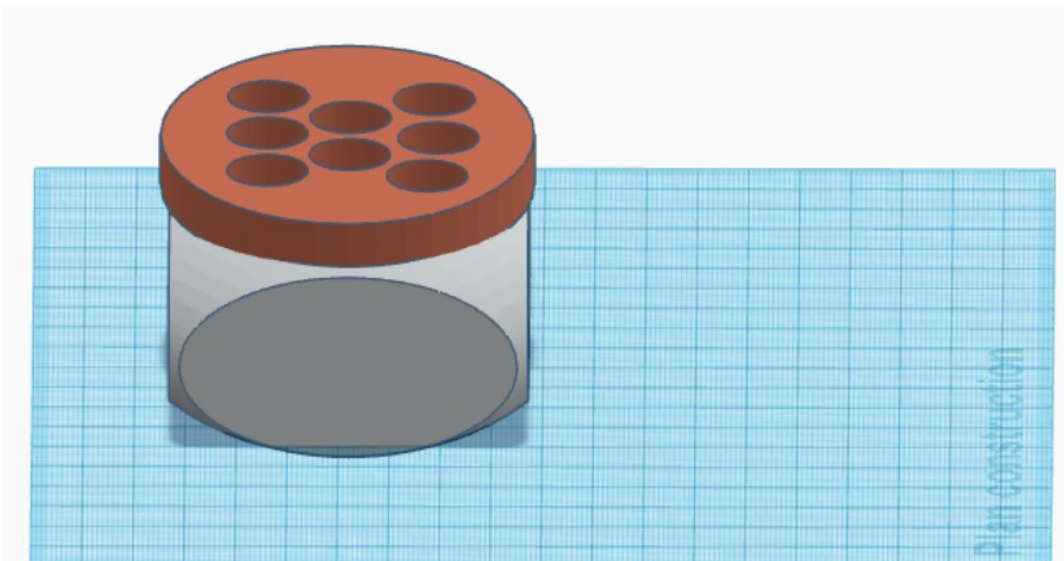
b. Créer la base du porte savon

Action : prendre la forme dé comme support

Dimensions : 90mm (longueur) x 74mm (largeur); et 40mm (épaisseur)



c. Juxtaposer le support et la grille



Action : s'aider de la commande plan

d. Créer le haut du tuyau pour la sortie d'eau (et pour que le savon ne se ramollisse pas)

Action : prendre la forme cylindre et aller dans ses paramètres afin de l'éditer au format perçage

Dimensions : 18mm x 17mm x 10mm (profondeur)

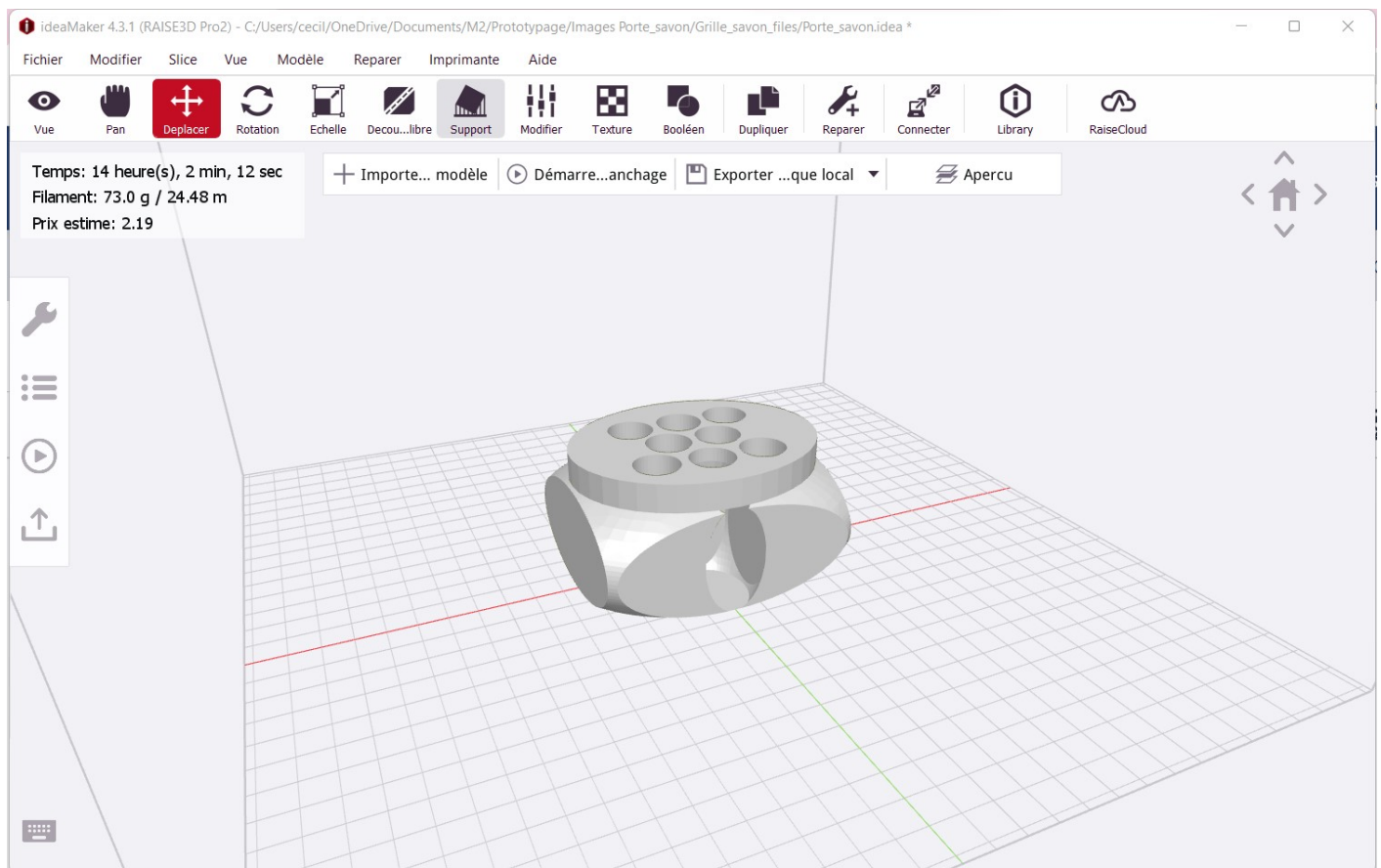
e. Créer la base du tuyau pour la sortie d'eau (et pour que le savon ne se ramollisse pas)

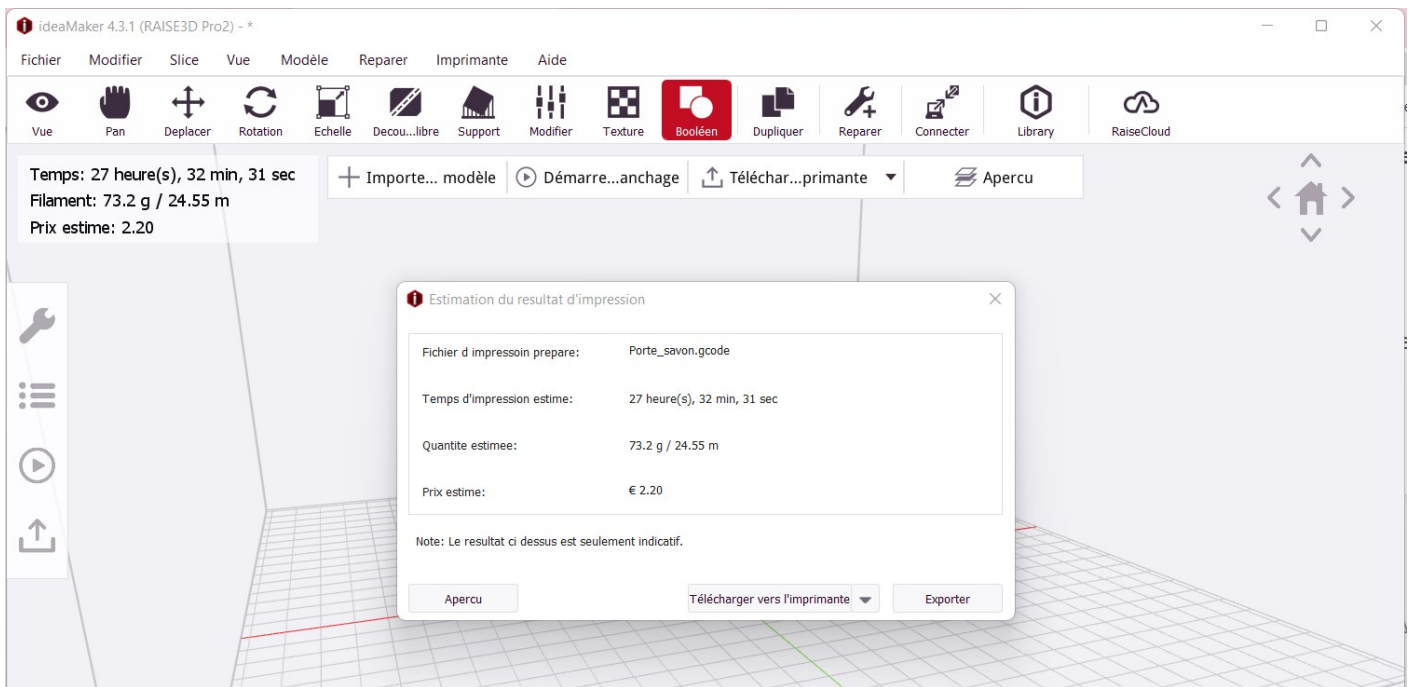
Action : prendre la forme dé et aller dans ses paramètres afin de l'éditer au format perçage

Dimensions : 30mm (longueur) x 22mm (largeur) x 39mm (profondeur)

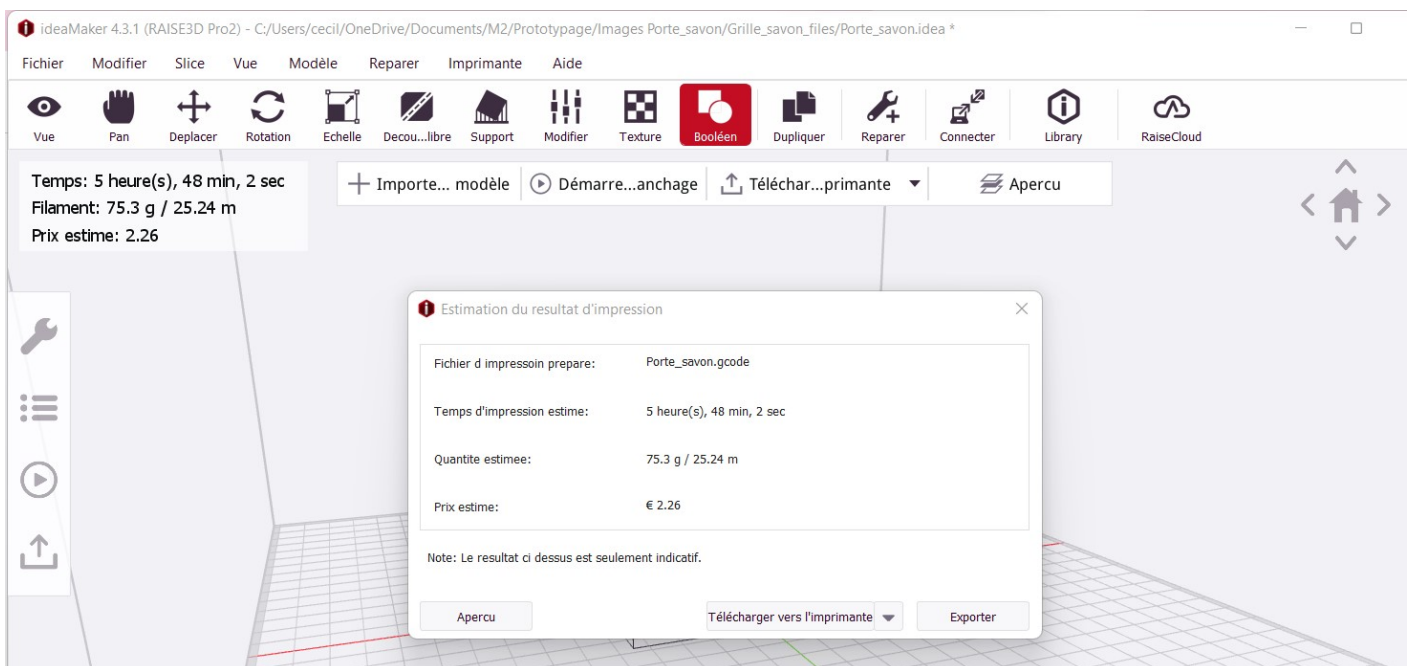
3.Impression 3D ultra haute qualité avec IdeaMaker

Toutes les impressions seront faites avec l'imprimante Raise 3D Pro2





4.Impression 3D rapide avec IdeaMaker

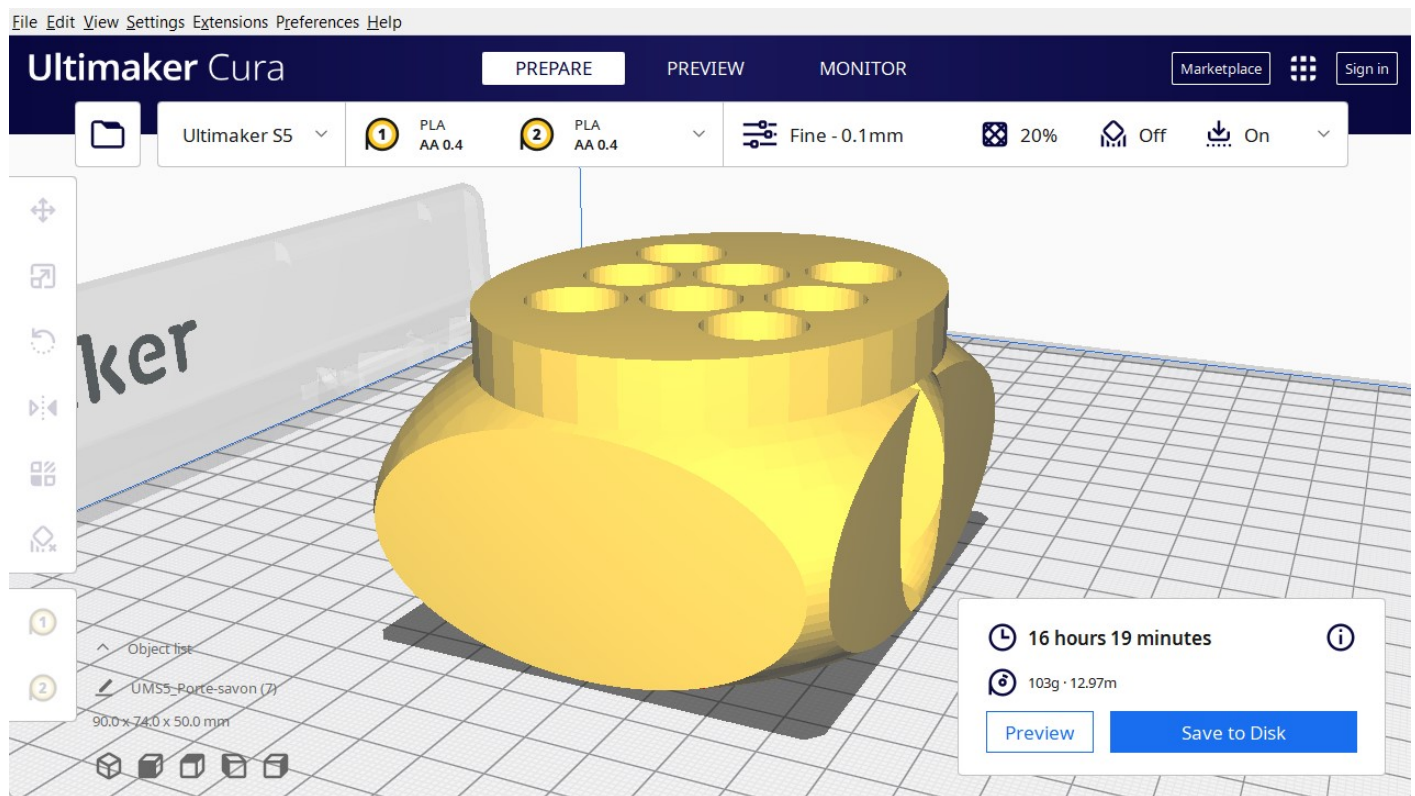


Fichier porte_savon.stl ci-joint

5.Impression 3D avec Ultimaker Cura avec l'imprimante Raise 3D Pro2 (simulation n°3)

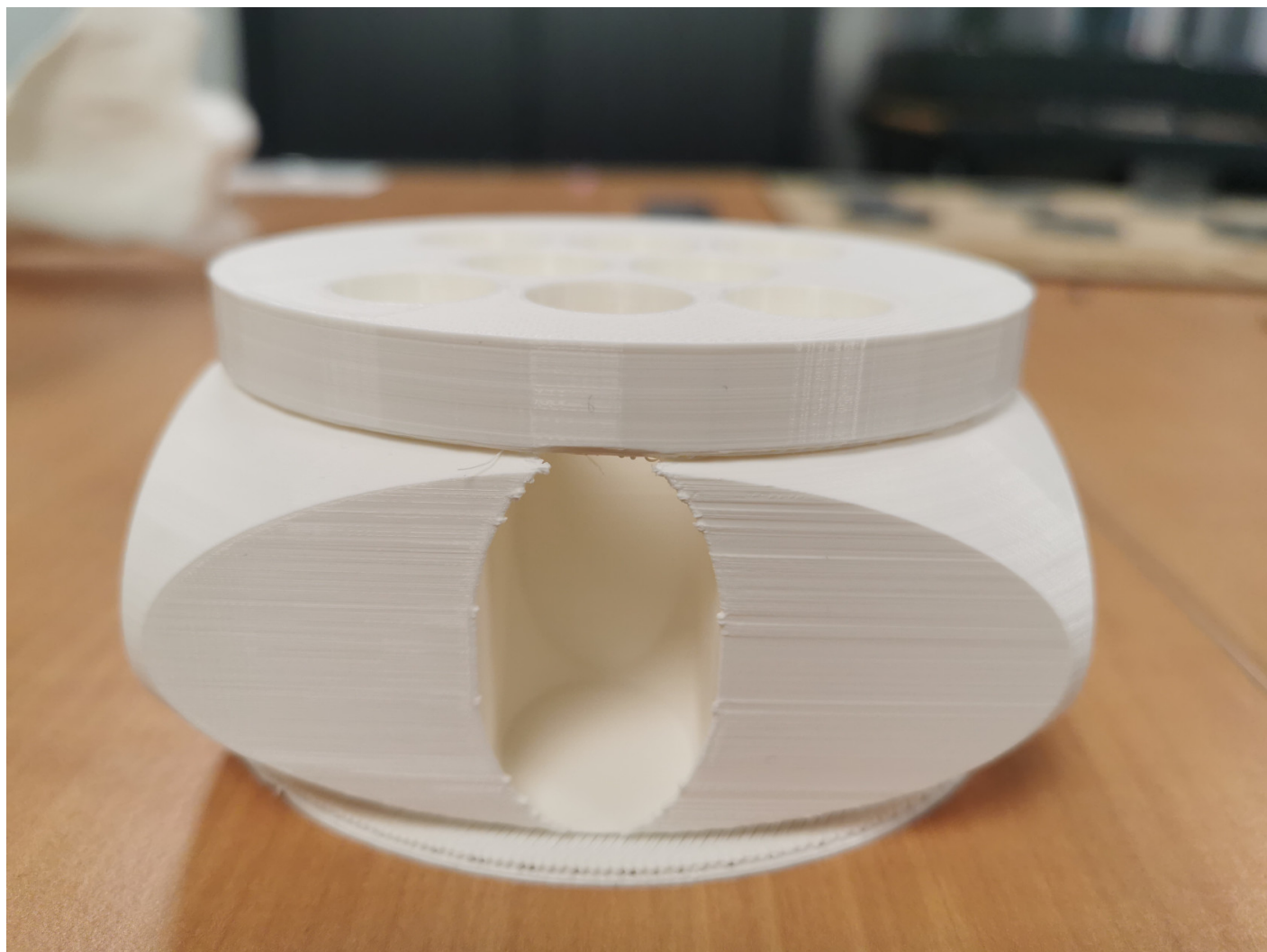
Actions :

- Importer le fichier porte_savon.stl
- Appuyer sur prepare
- Appuyer sur print



Fichier porte_savon.gcode non disponible car trop volumineux

6. Porte savon imprimé







Semaine 2 : projet montage électronique

Le but du projet est de réaliser un montage électronique avec une carte Arduino. Une carte Arduino est une carte électronique équipée d'un micro-contrôleur.

1. Modélisation de circuit électrique

Matériels :

- Carte Arduino

[Carte Arduino](#)

- Logiciel Tinkercad

[Tinkercad](#)

Loi d'Ohm (Loi de Base en électronique) : U (tension aux bornes de la résistance) = R (valeur de la résistance) * I (intensité du courant qui circule à travers la résistance)

A. Blink programme

Il s'agit d'un programme qui sert à allumer une LED pendant une seconde, puis l'éteint pendant une seconde, de manière répétée.

[Blink programme](#)

Code associé :

```
#define LED 10 // connecte la LED sur la pin 10
int compteur = 0;
void setup()
{
  Serial.begin(9600); // permet de dire que l'on va initier une communication entre
  l'ordinateur et le circuit à la vitesse 9600
  pinMode(LED, OUTPUT); // déclaration de la pin 10 comme sortie
}
void loop()
```

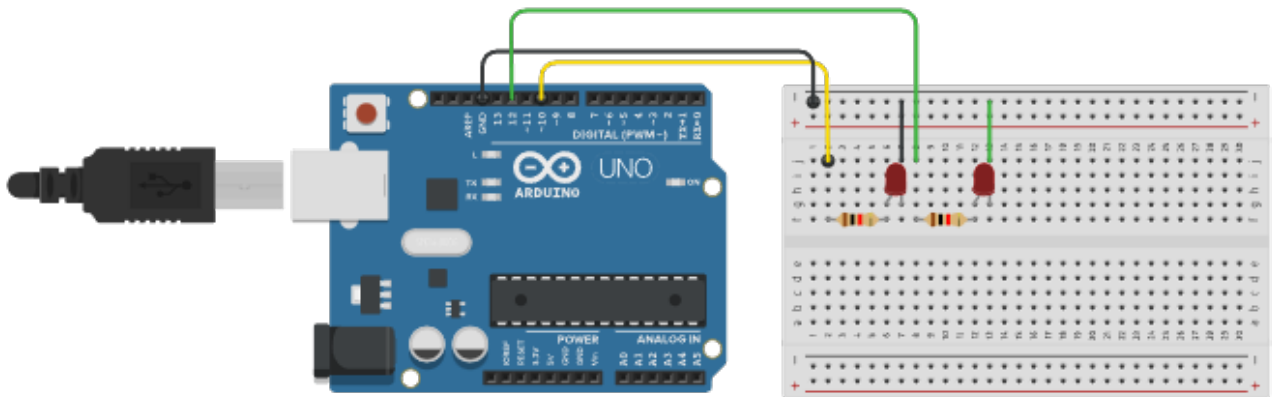
```
{  
Serial.print("Debut de la boucle ");  
Serial.println(compteur);  
digitalWrite(LED, HIGH); // Allume la LED  
delay(1000); // ne rien faire pendant 1000ms  
digitalWrite(LED, LOW);  
delay(1000); // ne rien faire pendant 1000ms  
compteur++; // incrémentation du compteur  
}
```

Bonus : on peut rajouter une deuxième LED

Code associé :

```
#define LEDA 10  
#define LEDB 12  
  
int compteur = 0;  
  
void setup()  
{  
  Serial.begin(9600);  
  
  pinMode(LEDA, OUTPUT); // déclaration de la pin 10 comme sortie  
  pinMode(LEDB, OUTPUT); // déclaration de la pin 10 comme sortie  
}  
  
void loop()  
{  
  Serial.print("Debut de la boucle ");  
  Serial.println(compteur);  
  
  digitalWrite(LEDA, HIGH);  
  digitalWrite(LEDB, LOW);  
  delay(1000); // ne rien faire pendant 1000ms  
  digitalWrite(LEDA, LOW);  
  digitalWrite(LEDB, HIGH);  
  delay(1000); // ne rien faire pendant 1000ms  
  
  compteur++; // incrémentation du compteur  
}
```

Image associé :



B. Bouton programme

Il s'agit d'un programme qui sert à allumer une LED via un bouton-poussoir.

Bouton programme

Code associé :

```
#define BTN 8
#define LED 10

void setup()
{
  pinMode(BTN, INPUT_PULLUP);
  pinMode(LED, OUTPUT); // déclaration de la pin 13 comme sortie

  Serial.begin(9600);
}

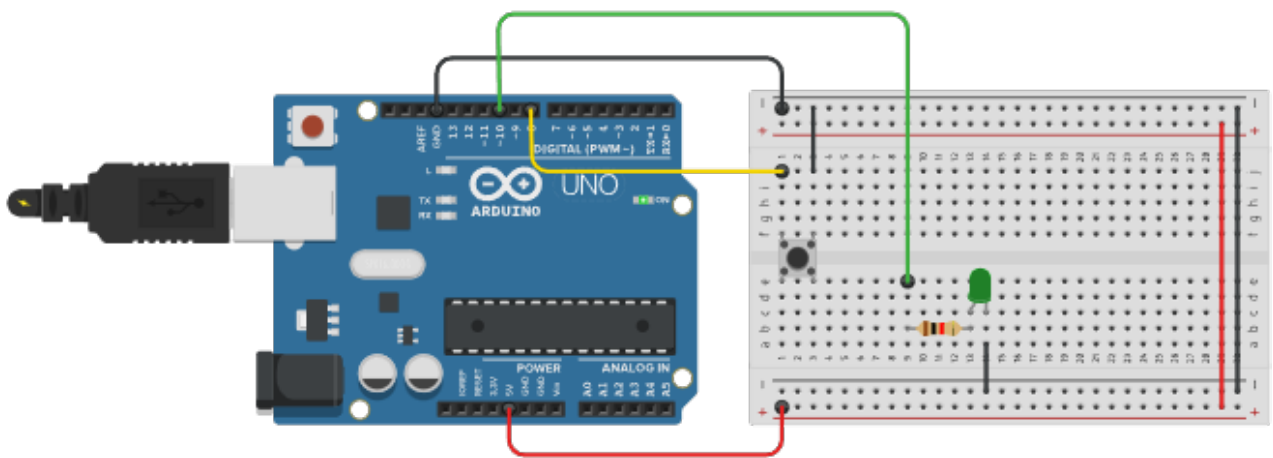
void loop()
{
```

```

if( digitalRead(BTN) == LOW )
{
  Serial.println("Button pressed");
  if( digitalRead(LED))
  {
    digitalWrite(LED, LOW);
  }
  else
  {
    digitalWrite(LED, HIGH);
  }
}
delay(100);
}

```

Image associé :



B. Buzzer programme

Il s'agit d'un programme qui sert à allumer le haut-parleur piézo via un bouton-poussoir.

[Buzzer programme](#)

Code associé :

```
#define POT A0
#define BUZZ 9
#define BTN 7

int etat = 0;

void setup()
{
  pinMode(POT, INPUT);
  pinMode(BUZZ, OUTPUT);
  pinMode (BTN, INPUT_PULLUP);

  Serial.begin(9600);
}

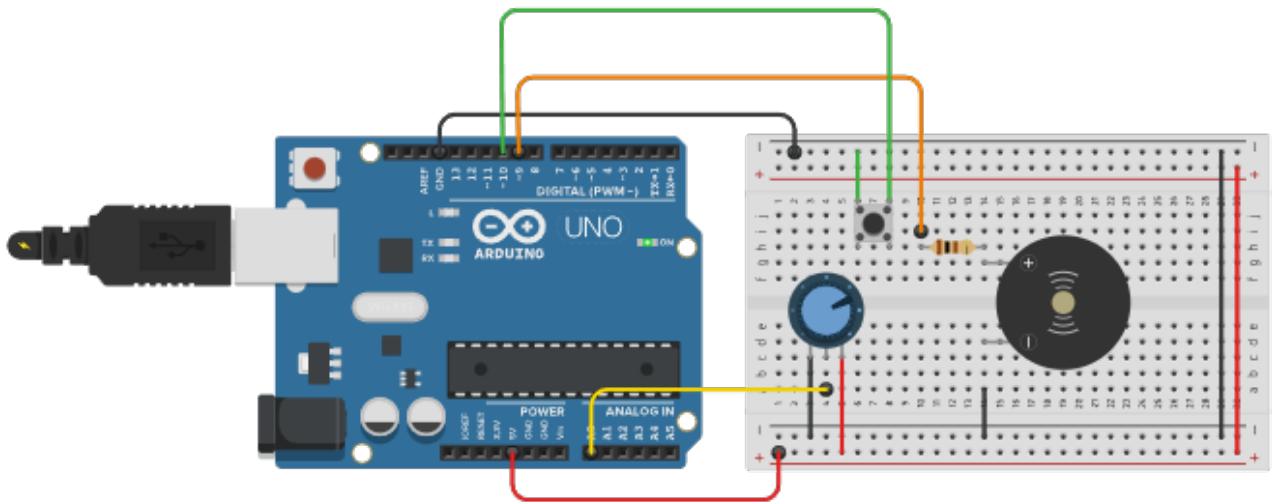
void loop()
{
  int pot = analogRead(POT);
  Serial.println(pot);

  int buzzfreq = map(pot, 0, 1023, 1000, 20000);

  if( digitalRead(BTN) == LOW )
  {
    if( etat == 1 )
    {
      noTone(BUZZ);
      etat = 0;
    }
    else
    {
      tone(BUZZ, buzzfreq);
      etat = 1;
    }
  }

  delay(120);
}
```

Image associé :



C. Capteur programme

Il s'agit d'un programme impliquant un capteur qui mesure la distance avec un LED qui s'allume si la distance est inférieure ou égale à 30m.

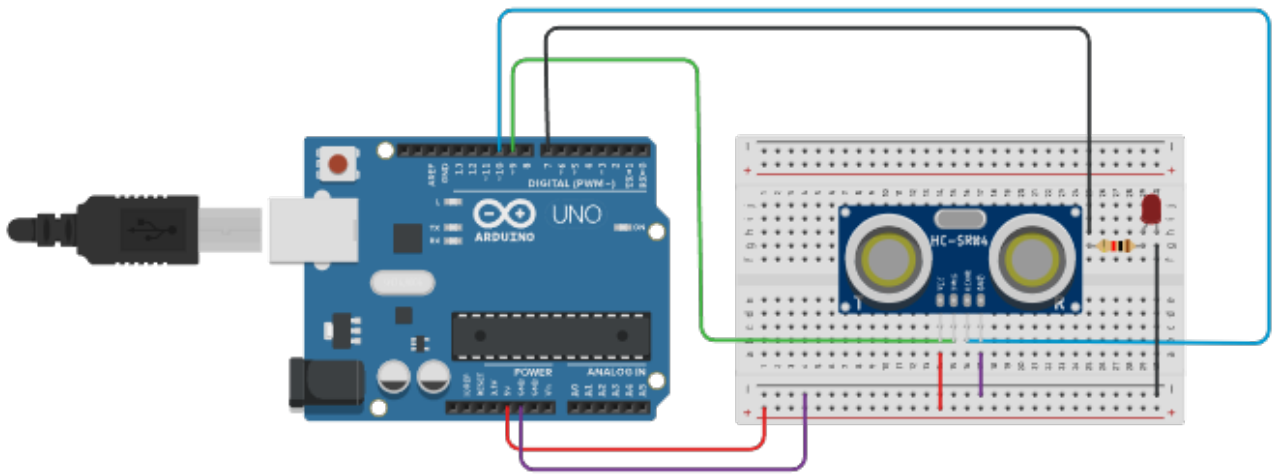
Capteur programme

```
/*  
  Ultrasonic Sensor HC-SR04 and Arduino Tutorial  
  
  */  
// defines pins numbers  
const int trigPin = 9;  
const int echoPin = 10;  
#define LED 7  
// defines variables  
long duration;  
int distance;  
void setup() {  
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output  
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
```

```
pinMode(LED, OUTPUT);
Serial.begin(9600); // Starts the serial communication
}
void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);

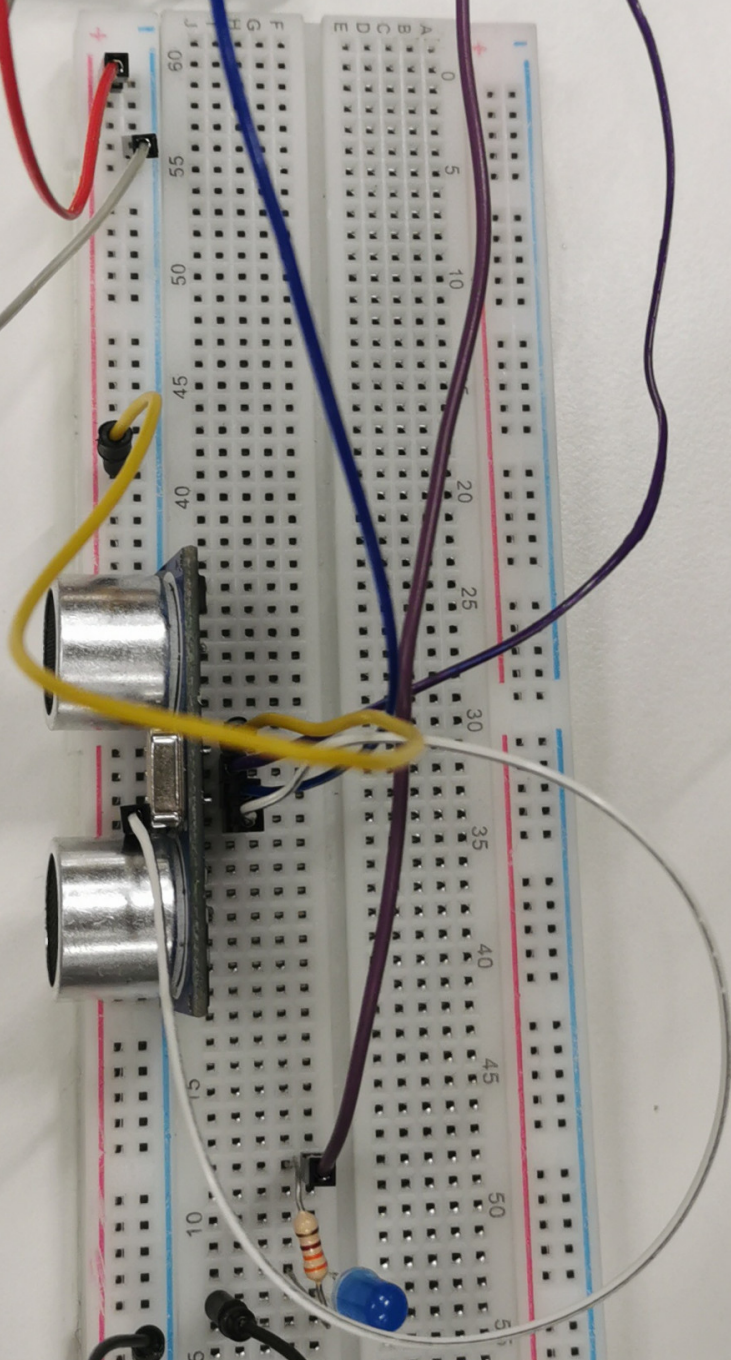
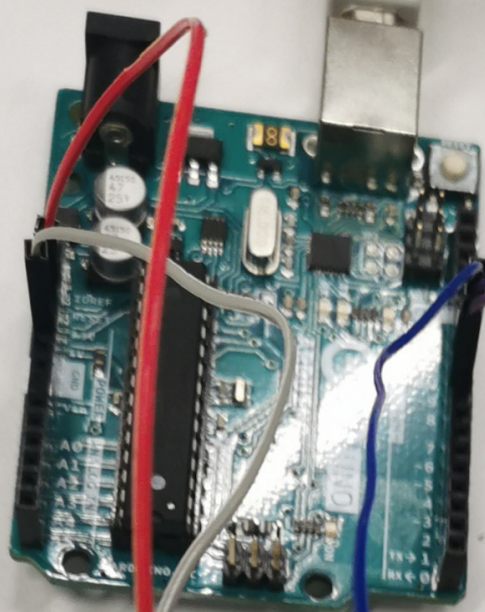
  if (distance <= 30)
  {
    digitalWrite(LED, HIGH);
  }
  else
  {
    digitalWrite(LED, LOW);
  }
}
```

Image associé :

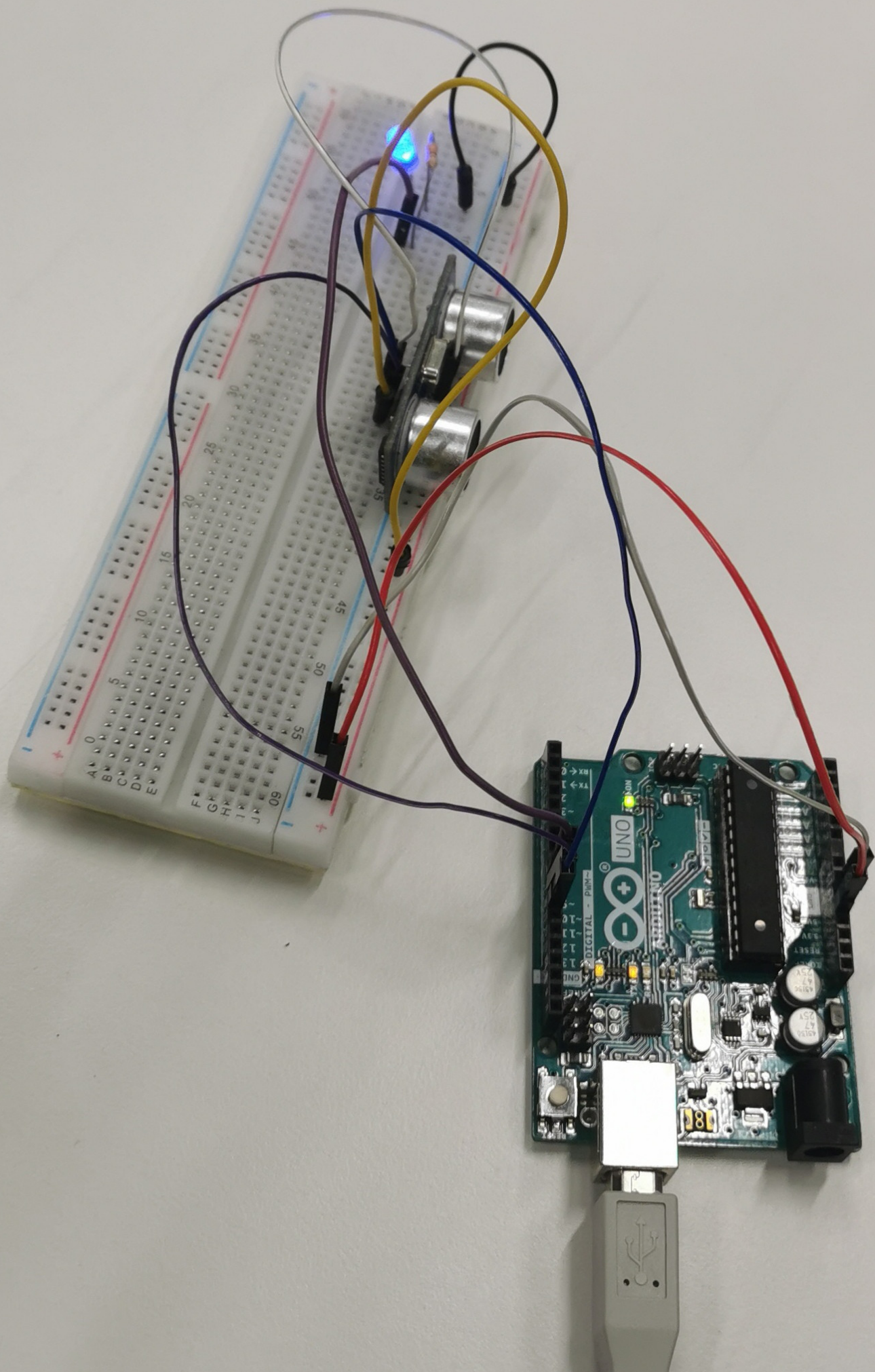


Réalisations associé :

Plus de 30m :



Moins ou égale à 30m :



Il existe également pleins d'autres programmes à réaliser avec la carte Arduino (Moteur, Capteur avec Action Moteur, Capteur de Temperature, ...)

[Moteur programme](#)

[Temperature capteur programme](#)

Revision #17

Created 14 February 2023 10:45:29 by Dieval Cecile Lina

Updated 1 May 2023 20:21:39 by Dieval Cecile Lina