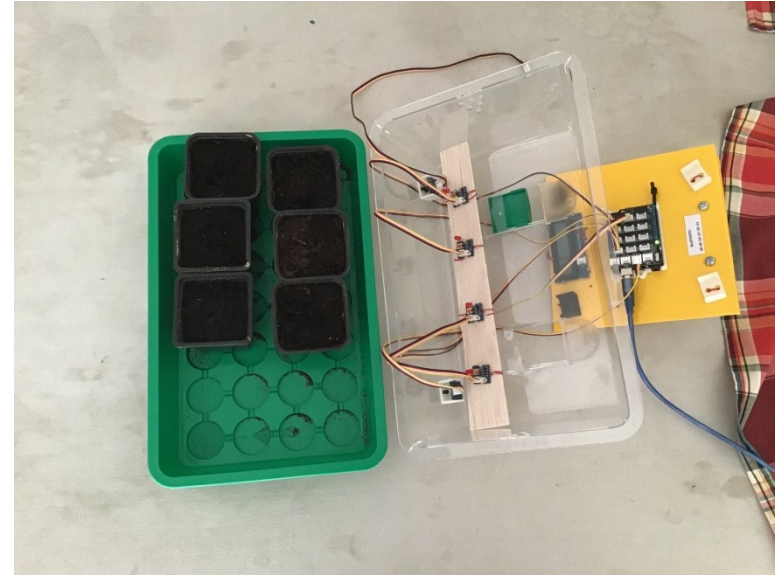
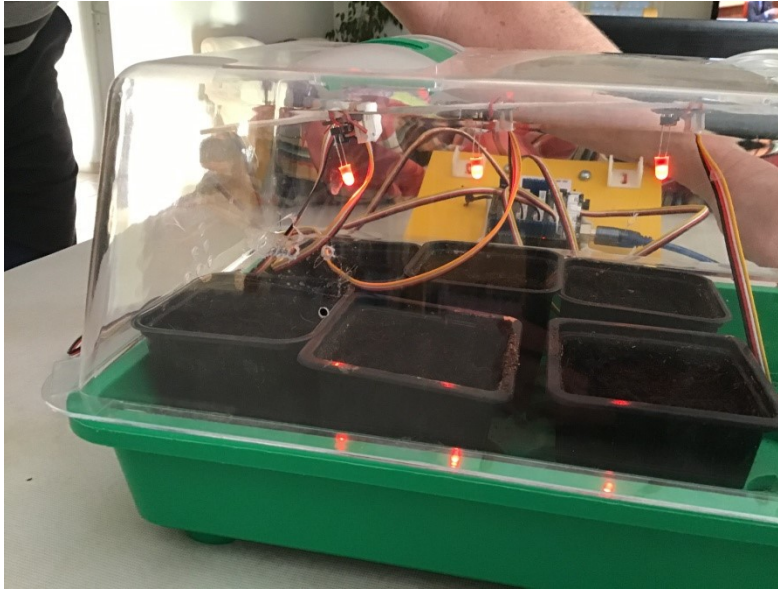


SCENARIO 2 – PROJET 2

ECLAIRAGE AUTOMATIQUE



Fonctionnement du système :

Si le capteur de lumière branché en A1 est en dessous de 400 alors il faut mettre en marche l'allumage de la serre.
Cet allumage se fait par le relais d'alimentation 220 volts branché en D2 D3 D4 OU D5 qui pilote les lampes ROUGE.

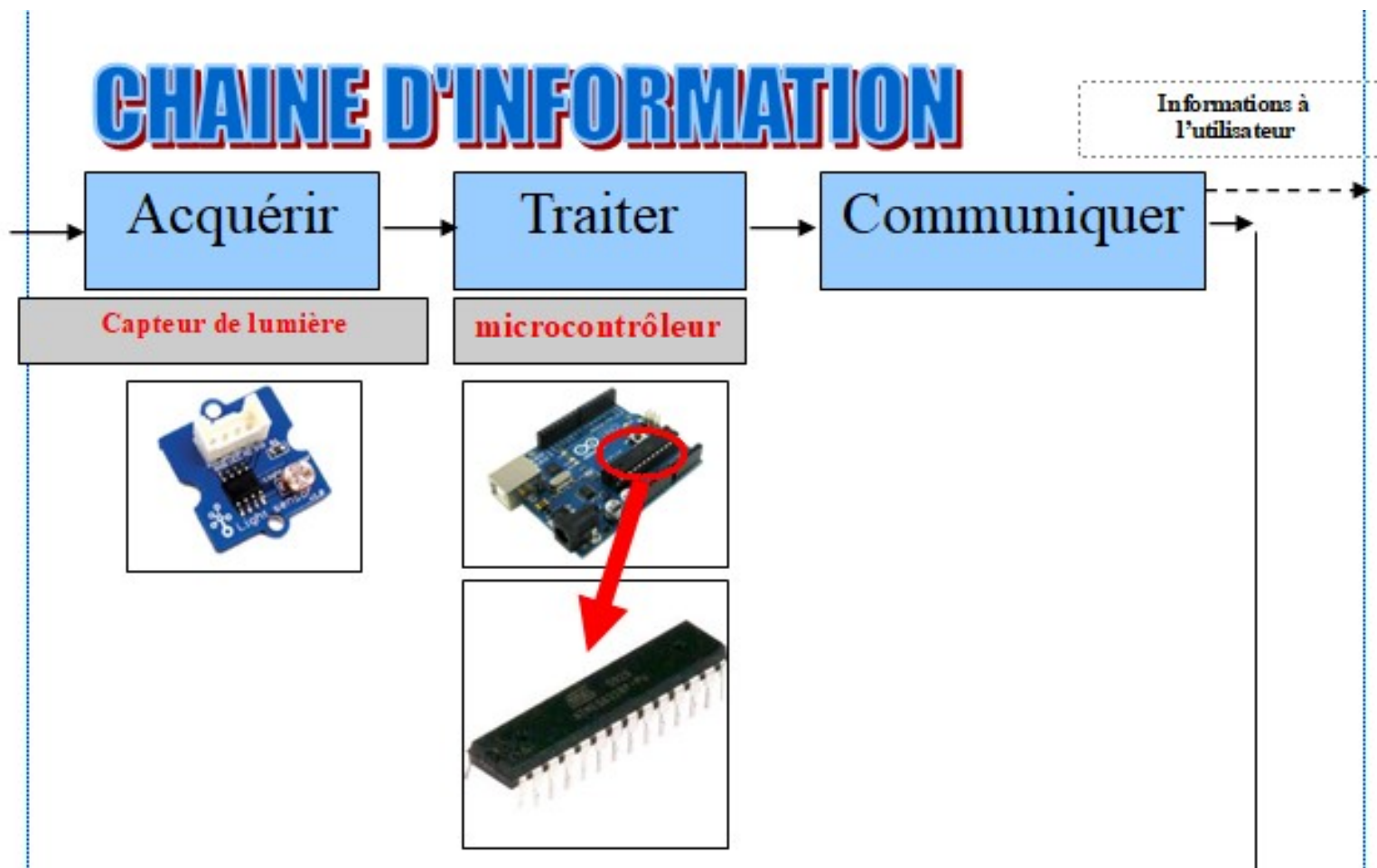
LA SERRE DE JARDIN



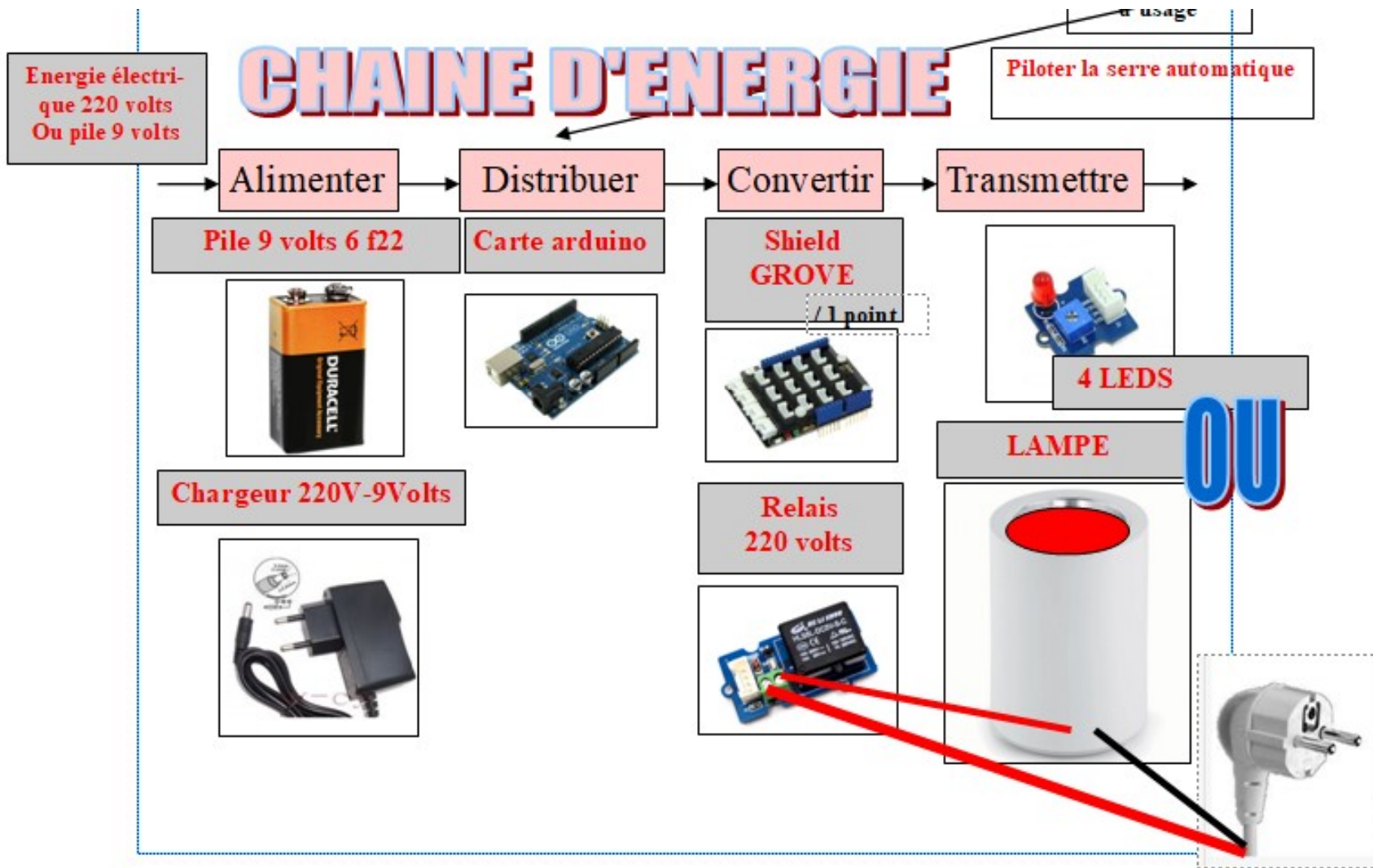
Fonctionnement du système :

Les lumières rouge sont utilisées pour la croissance des plantes la nuit, ou pour la croissance des tomates.

CHAINE D'INFORMATION ET D'ENERGIE DE LA SERRE :



LA CHAINE D'ENERGIE



LE CAPTEUR DE LUMIERE

Capteur de lumière Grove 101020132



 seeed

Détecteur de lumière Grove V1.2 **101020132**

Code article : **34782**

Ce capteur de lumière compatible Grove permet de détecter la présence de lumière. La tension de sortie analogique évolue de 0 à +Vcc suivant l'intensité lumineuse mesurée.

[Description complète](#)

Quantité : + -

✓ Quantité en stock : 820

 Livraison à partir de 2,90€ 

2,54 € HT

3,05 € TTC

dont 0,02 € d'éco-part

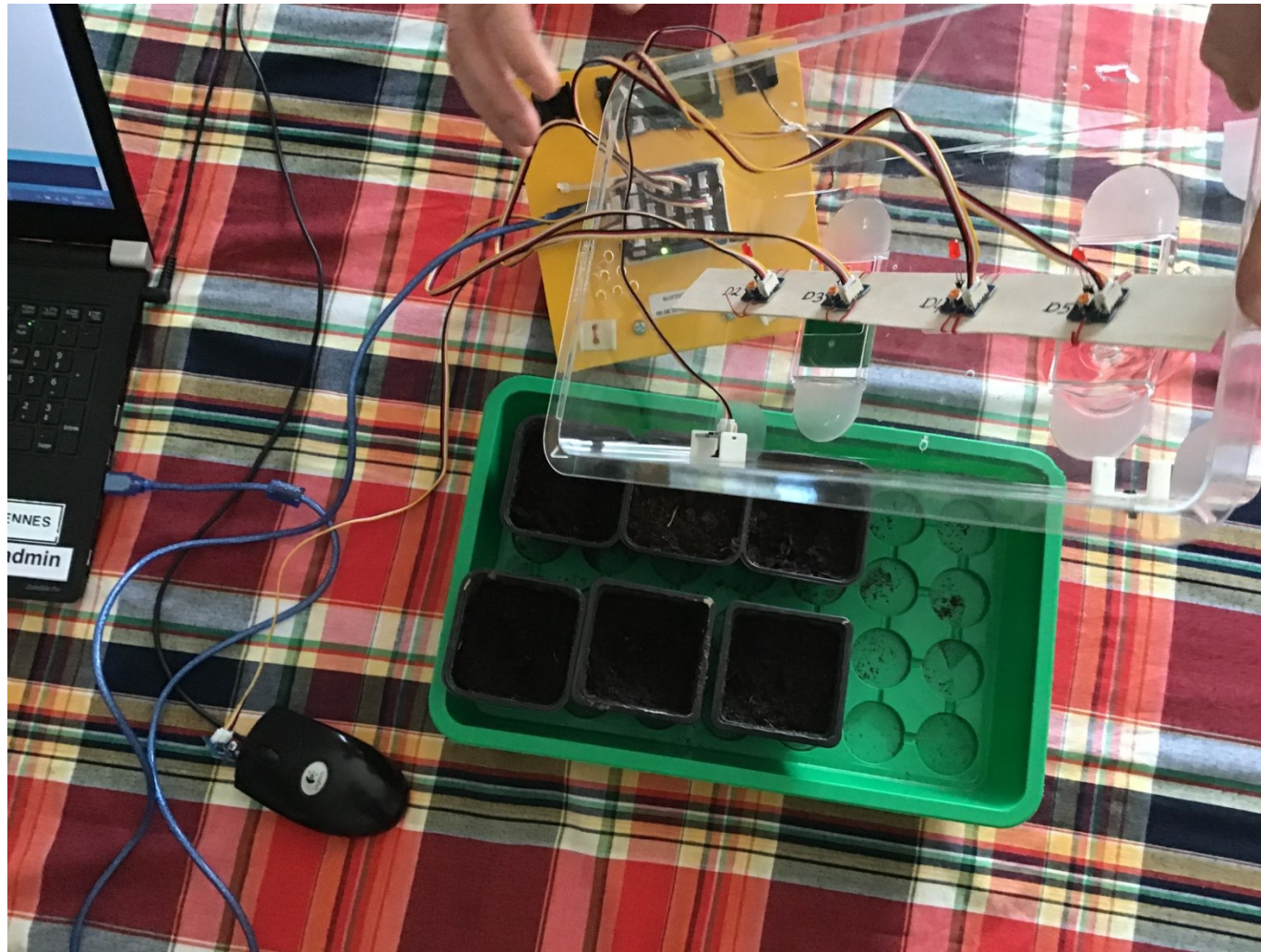
LES DONNEES DU CAPTEUR

- - la grandeur à mesurer : **la lumière en lux**
- - la plage de mesure de mesure du capteur : **????**
- - la résolution (précision du capteur) : **????**
- - le type de signal fourni par le capteur : **analogique de 0 à 1024**
- - la tension d'alimentation : **0 à 3.3 ou 5V**
- Les informations du capteur ci-dessous :

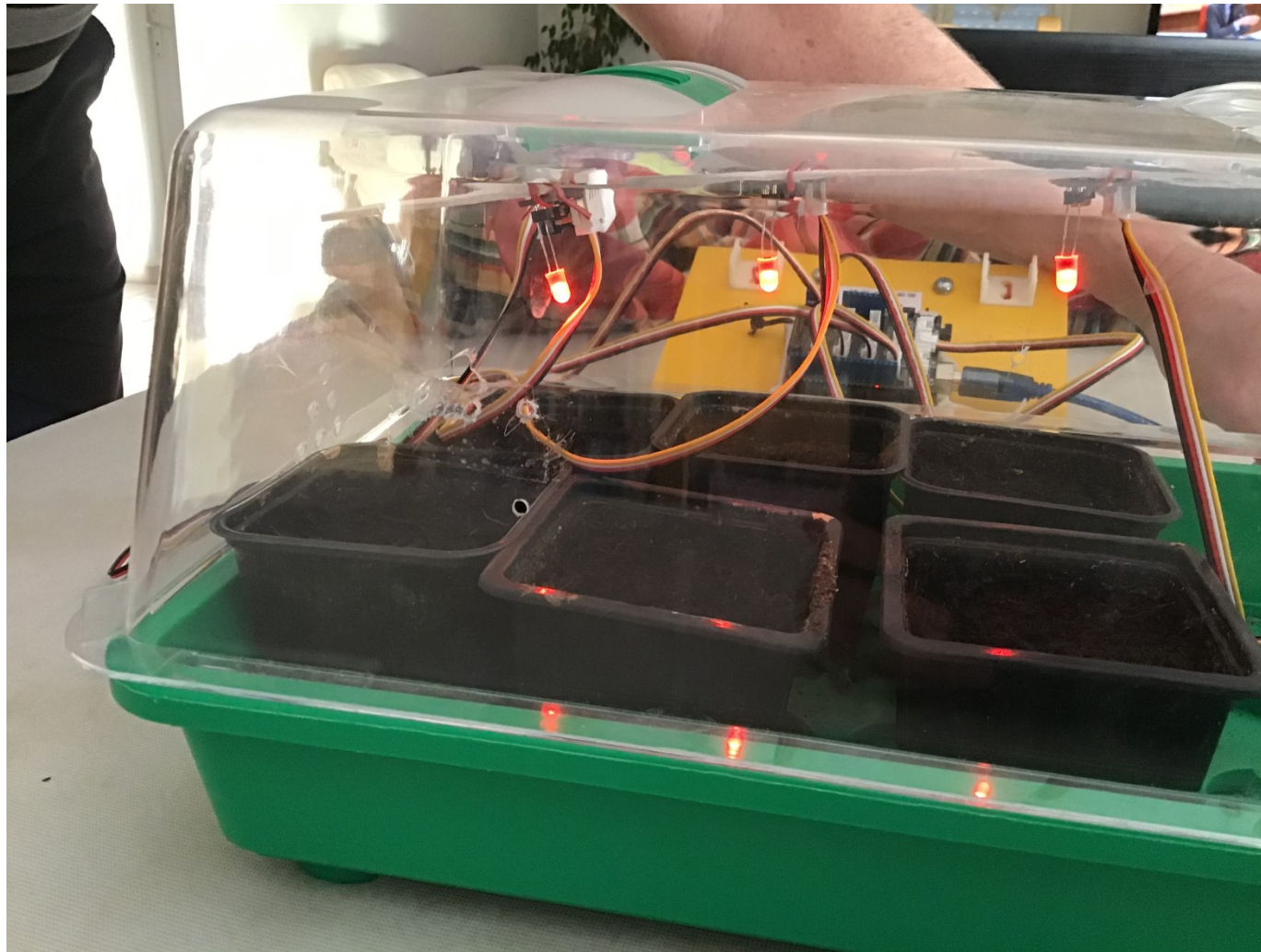
http://wiki.seeedstudio.com/Grove-Light_Sensor/

<https://www.gotronic.fr/art-detecteur-de-lumiere-grove-v1-2-101020132-25427.htm>

LE MONTAGE



LE MONTAGE



LE MONTAGE



LE PROGRAMME DE MESURE DE LUMIERE

```
// SERRE DE TOMATES - ECLAIRAGE DURANT LA NUIT AVEC LUMIERE ROUGE  
// PERENNES CLEMENT- 1ERE STI2D
```

```
// Defines the pins to which the light sensor and LED are connected.  
const int ldr_lumiere = A1;  
const int led2 = 2;  
const int led3 = 3;  
const int led4 = 4;  
const int led5 = 5;
```

```
int seuil_jour_nuit = 500;
```

```
void setup()  
{  
  pinMode(led2, OUTPUT);  
  pinMode(led3, OUTPUT);  
  pinMode(led4, OUTPUT);  
  pinMode(led5, OUTPUT);  
}
```

```
void loop()  
{  
  
  int tauxlumiere = analogRead(ldr_lumiere);  
  
  if(tauxlumiere < seuil_jour_nuit)  
  {  
    digitalWrite(led2, HIGH);  
    digitalWrite(led3, HIGH);  
    digitalWrite(led4, HIGH);  
    digitalWrite(led5, HIGH);  
  
  }  
  else  
  {  
    digitalWrite(led2, LOW);  
    digitalWrite(led3, LOW);  
    digitalWrite(led4, LOW);  
    digitalWrite(led5, LOW);  
  }  
}
```

SIMULATION THINKERCARD < 400

à nuit

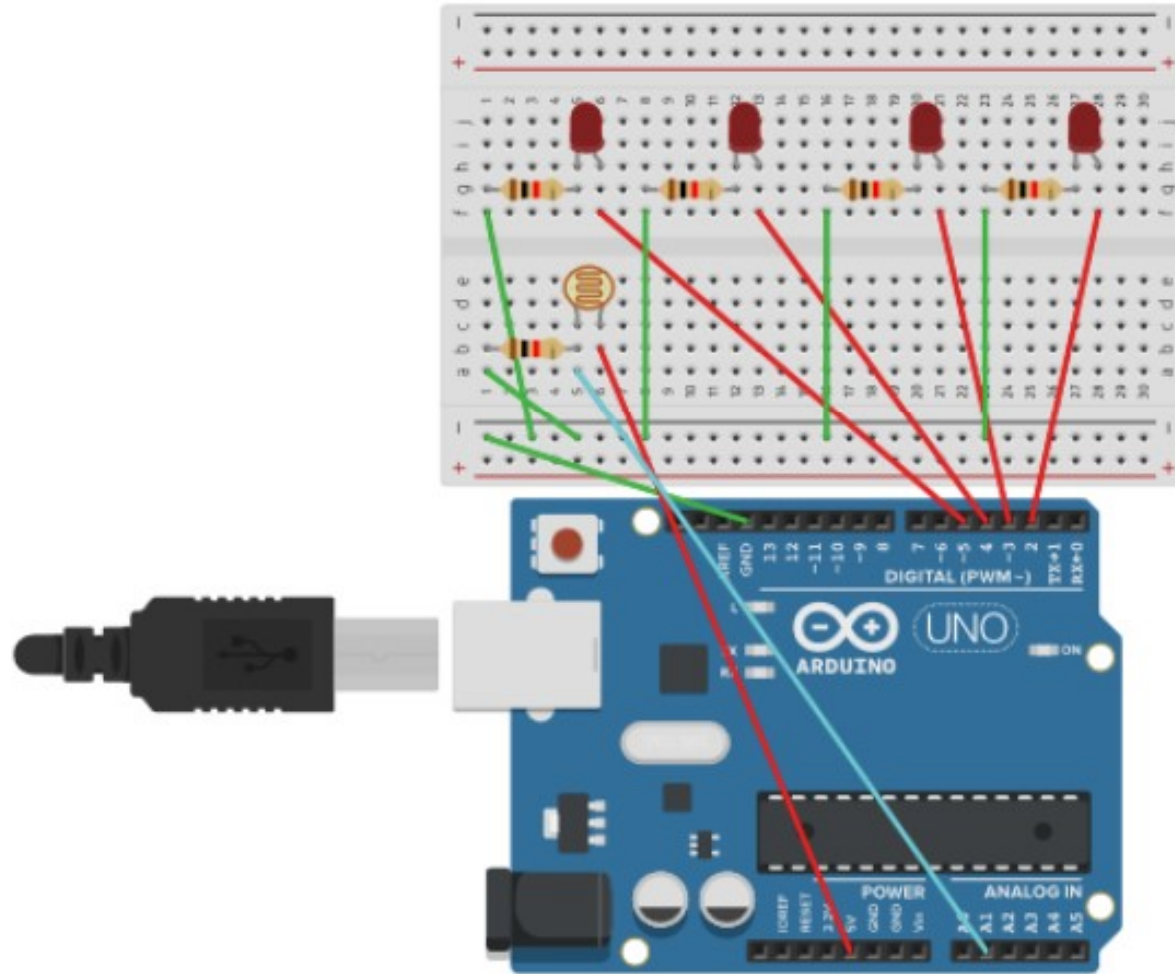
toutes les modifications ont été enregistrées

The screenshot displays the Tinkercard simulation environment. At the top, the text "à nuit" is on the left and "toutes les modifications ont été enregistrées" is on the right. Below this, a status bar shows "Durée du simulateur: 00:00:01.873" and two buttons: "Code" and "Arrêter la simulation".

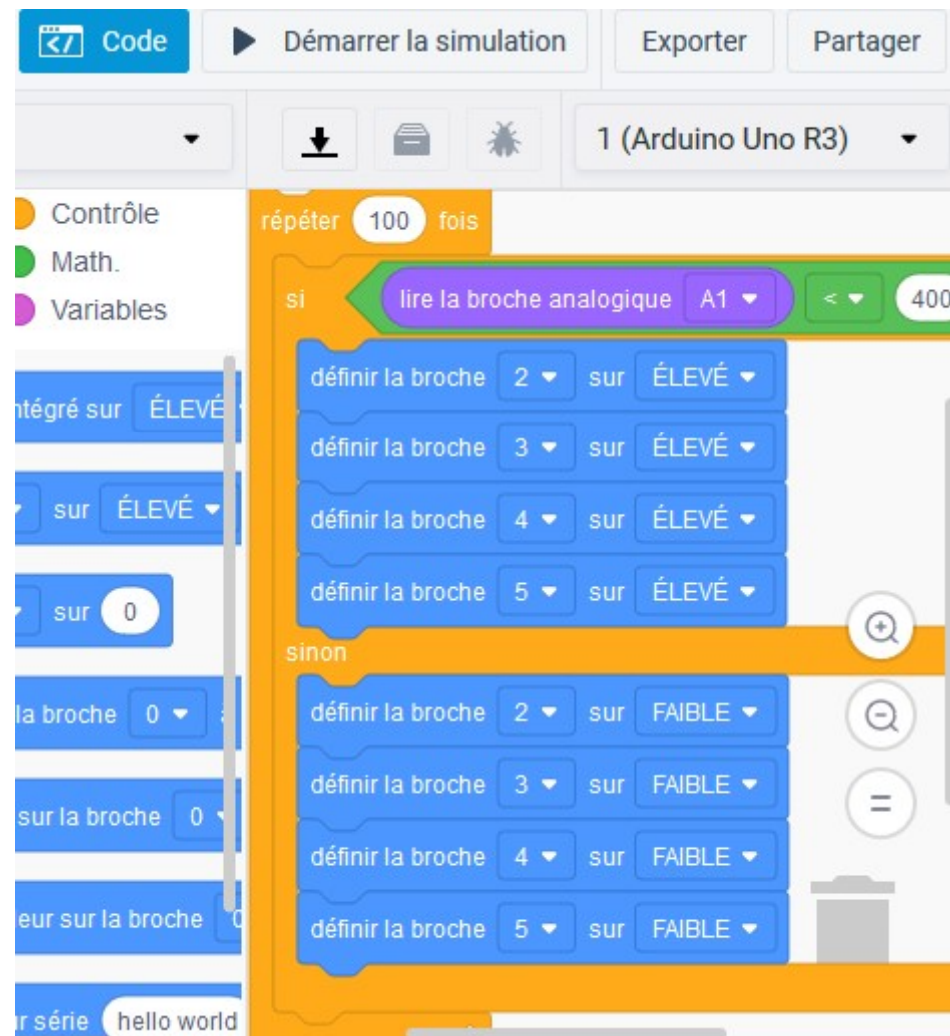
The main workspace features an Arduino Uno board connected to a breadboard. The breadboard contains a photoresistor, four resistors, and four LEDs. Wires connect the components to the Arduino's digital pins. A blue box highlights the "Photorésistance" component in the component palette, with its name set to "lumiere".

On the right side, the component palette is visible, showing "Composants De base" and a search bar labeled "Rechercher". Below the search bar, there are icons for a resistor (labeled "Résistanc") and a push button (labeled "Bouton pous").

SIMULATION THINKERCARD > 400



PROGRAMME SCRATCH



PROGRAMME ARDUINO

```
int counter;

void setup()
{
  pinMode(A1, INPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop()
{
  for (counter = 0; counter < 100; ++counter) {
    if (analogRead(A1) < 400) {
      digitalWrite(2, HIGH);
      digitalWrite(3, HIGH);
      digitalWrite(4, HIGH);
      digitalWrite(5, HIGH);
    } else {
      digitalWrite(2, LOW);
      digitalWrite(3, LOW);
      digitalWrite(4, LOW);
      digitalWrite(5, LOW);
    }
  }
  delay(10); // Delay a little bit to improve simulation performance
}
```