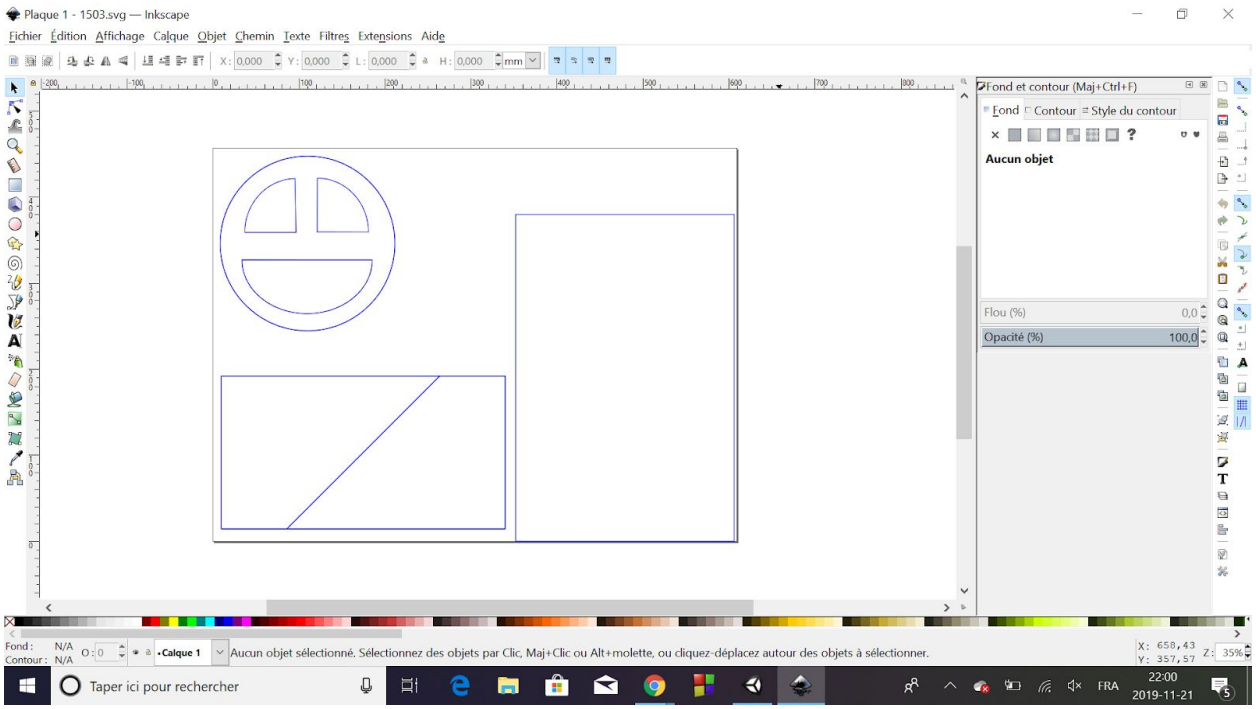
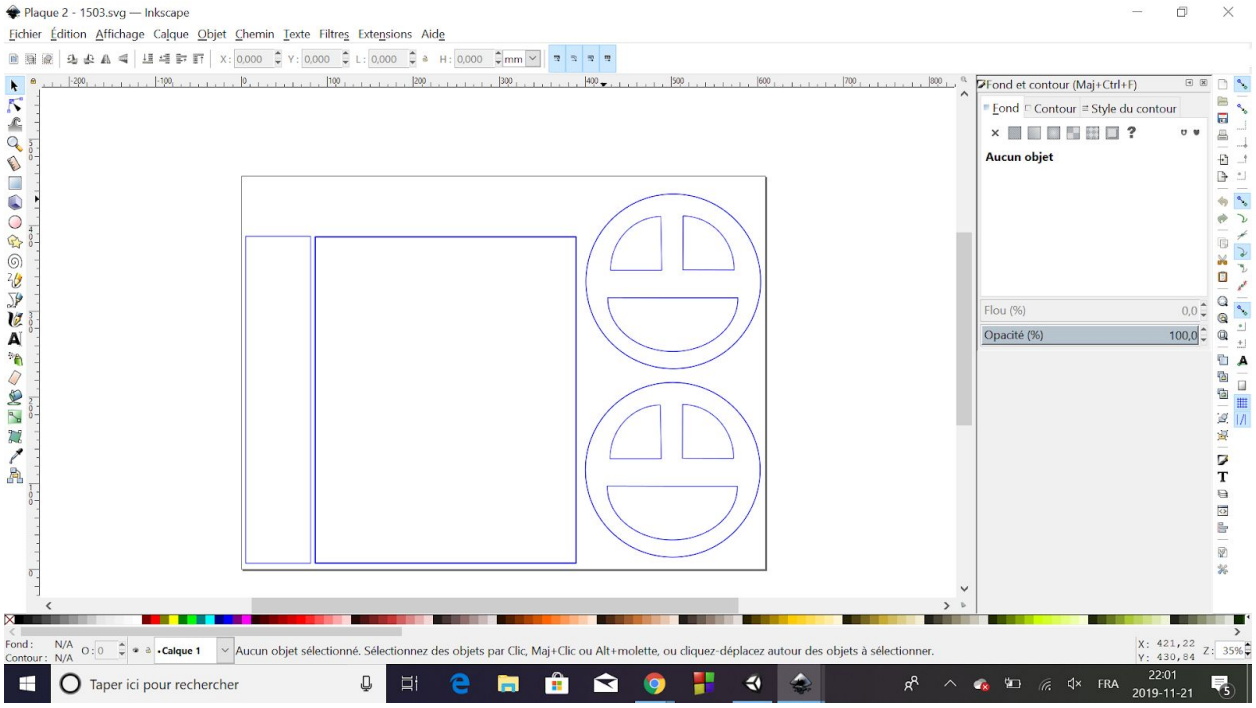
Logiciel Inkscape pour le découpage au laser





Le Code Arduino utilisé

Avant d'utiliser ce code, il faut télécharger le fichier Unojoy.h et l'inclure dans le même fichier que le code pour pouvoir l'exécuter.

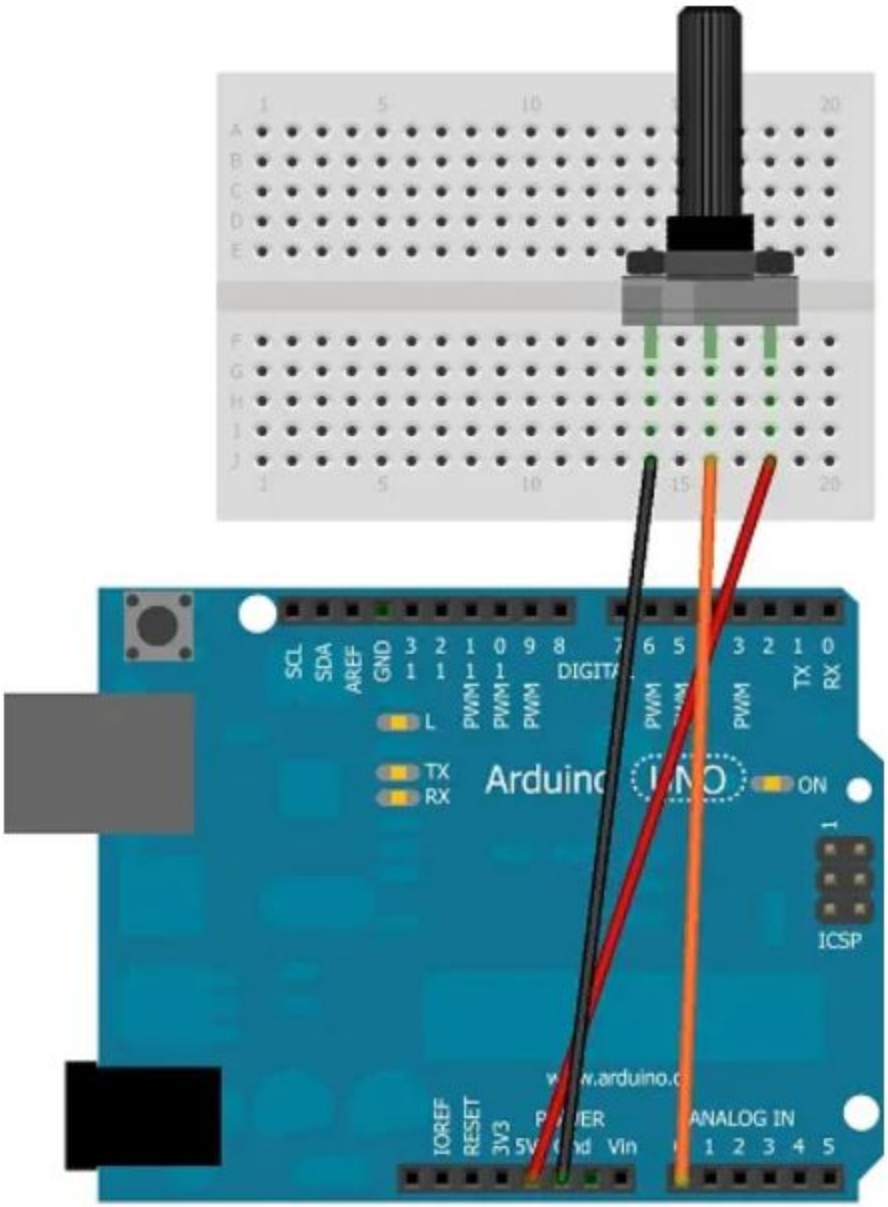
UnoJoyArduinoSample | UnoJoy.h

```
#include "UnoJoy.h"

void setup(){
  setupPins();
  setupUnoJoy();
}

void loop(){
  // Always be getting fresh data
  dataForController_t controllerData = getControllerData();
  setControllerData(controllerData);
}

void setupPins(void){
  // Set all the digital pins as inputs
  // with the pull-up enabled, except for the
  // two serial line pins
  for (int i = 2; i <= 12; i++){
    pinMode(i, INPUT);
    digitalWrite(i, HIGH);
  }
  pinMode(A4, INPUT);
  digitalWrite(A4, HIGH);
  pinMode(A5, INPUT);
  digitalWrite(A5, HIGH);
}

dataForController_t getControllerData(void){

  // Set up a place for our controller data
  //  Use the getBlankDataForController() function, since
  //  just declaring a fresh dataForController_t tends
  //  to get you one filled with junk from other, random
  //  values that were in those memory locations before
  dataForController_t controllerData = getBlankDataForController();
  // Since our buttons are all held high and
  //  pulled low when pressed, we use the "!"
  //  operator to invert the readings from the pins
  controllerData.triangleOn = !digitalRead(2);
  controllerData.circleOn = !digitalRead(3);
  controllerData.squareOn = !digitalRead(4);
  controllerData.crossOn = !digitalRead(5);
  controllerData.dpadUpOn = !digitalRead(6);
```

UnoJoyArduinoSample § | UnoJoy.h

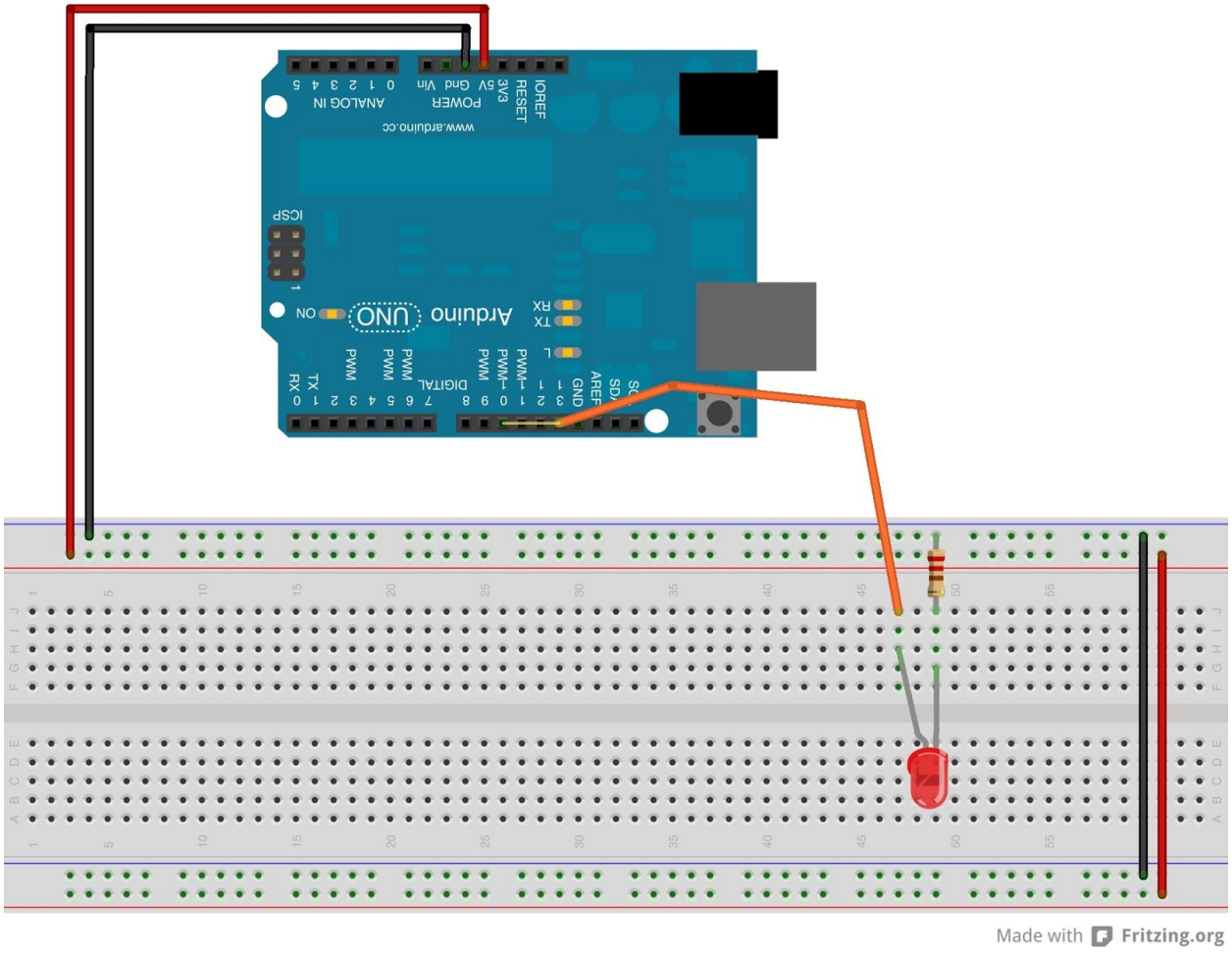
```
    pinMode(i, INPUT);
    digitalWrite(i, HIGH);
  }
  pinMode(A4, INPUT);
  digitalWrite(A4, HIGH);
  pinMode(A5, INPUT);
  digitalWrite(A5, HIGH);
}

dataForController_t getControllerData(void){

  // Set up a place for our controller data
  //  Use the getBlankDataForController() function since
  //  just declaring a fresh dataForController_t tends
  //  to get you one filled with junk from other, random
  //  values that were in those memory locations before
  dataForController_t controllerData = getBlankDataForController();
  // Since our buttons are all held high and
  //  pulled low when pressed, we use the "!"
  //  operator to invert the readings from the pins
  controllerData.triangleOn = !digitalRead(2);
  controllerData.circleOn = !digitalRead(3);
  controllerData.squareOn = !digitalRead(4);
  controllerData.crossOn = !digitalRead(5);
  controllerData.dpadUpOn = !digitalRead(6);
  controllerData.dpadDownOn = !digitalRead(7);
  controllerData.dpadLeftOn = !digitalRead(8);
  controllerData.dpadRightOn = !digitalRead(9);
  controllerData.l1On = !digitalRead(10);
  controllerData.r1On = !digitalRead(11);
  controllerData.selectOn = !digitalRead(12);
  controllerData.startOn = !digitalRead(A4);
  controllerData.homeOn = !digitalRead(A5);

  // Set the analog sticks
  //  Since analogRead(pin) returns a 10 bit value,
  //  we need to perform a bit shift operation to
  //  lose the 2 least significant bits and get an
  //  8 bit number that we can use
  controllerData.leftStickX = analogRead(A0) >> 2;

  // And return the data!
  return controllerData;
}
```

Branchement du potentiomètre



Potentiomètre ----------	Arduino
PIN 1	5V
PIN 2	A0
PIN 3	Gnd

Branchement des lumières LED



On fait ce branchement pour chaque lumière et on ajoute les boutons. Il suffit que ça soit connecté aux deux pins de la led.

Produit final