Arduino and Processing

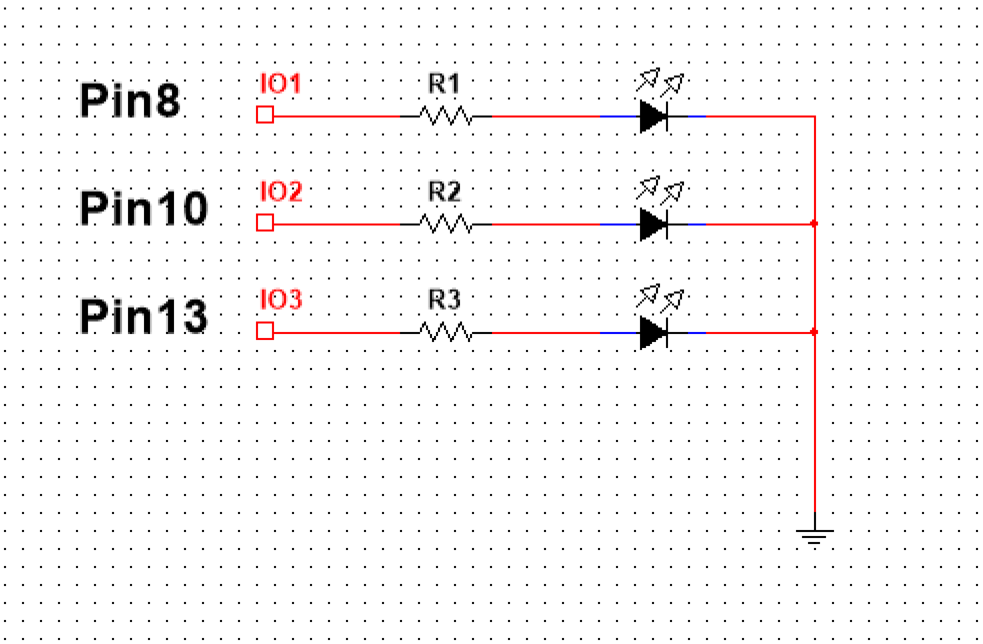
[POJET FINAL(EXAMEN) D’ELECTRONIQUE]

Alain MUSONI

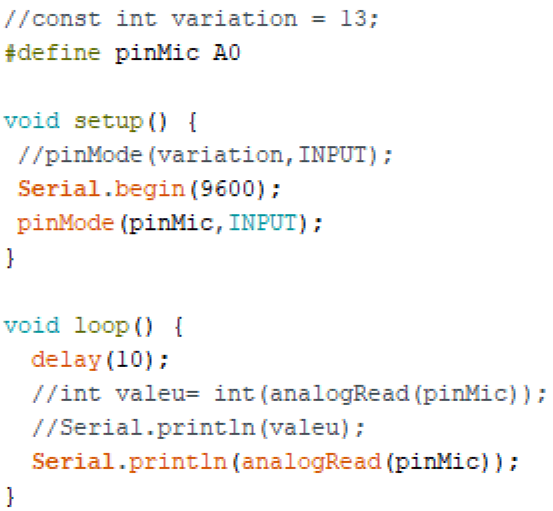
HEL | IS | 2017-2018

1. Tester La connexion avec des LEDs en utilisant Processing controls

- Branchement :

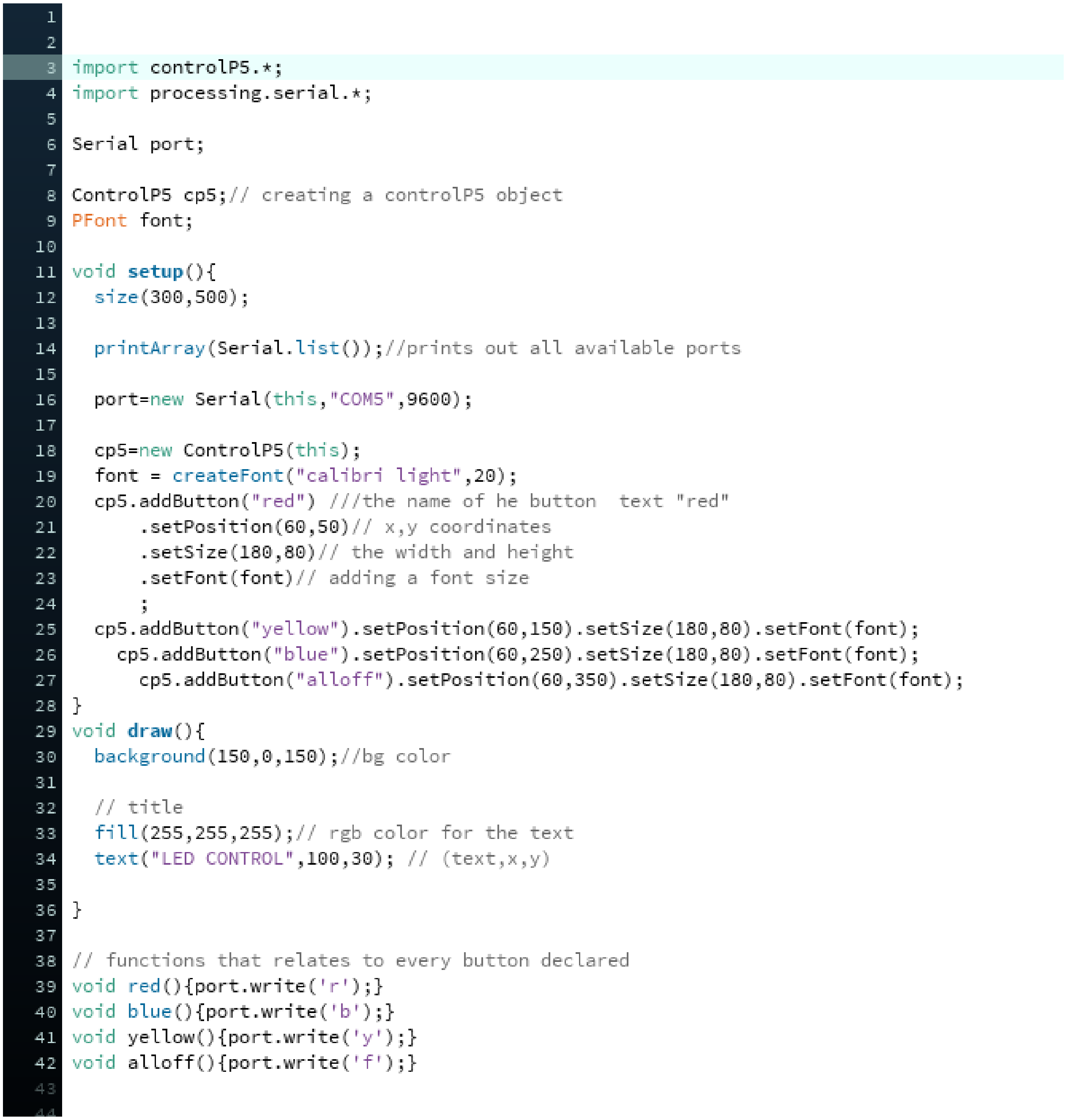


- Code Arduino :



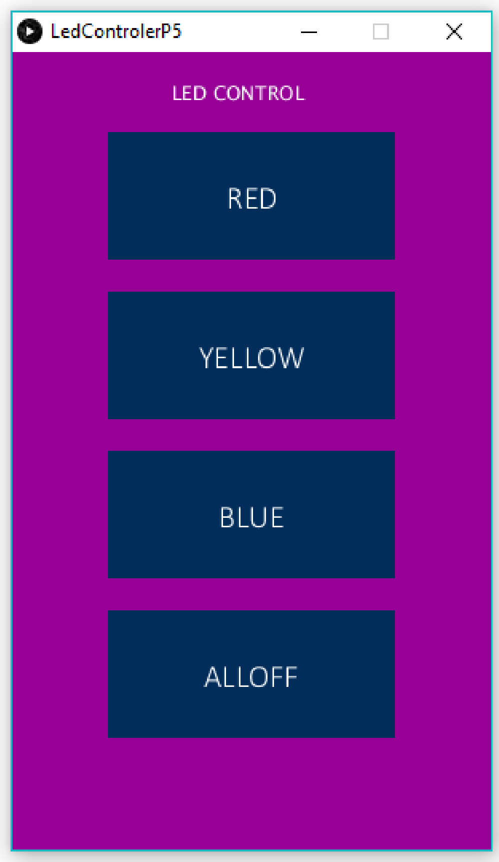
```
//const int variation = 13;
#define pinMic A0

void setup() {
 //pinMode(variation,INPUT);
 Serial.begin(9600);
 pinMode(pinMic,INPUT);
}

void loop() {
  delay(10);
  //int valeu= int(analogRead(pinMic));
  //Serial.println(valeu);
  Serial.println(analogRead(pinMic));
}
```

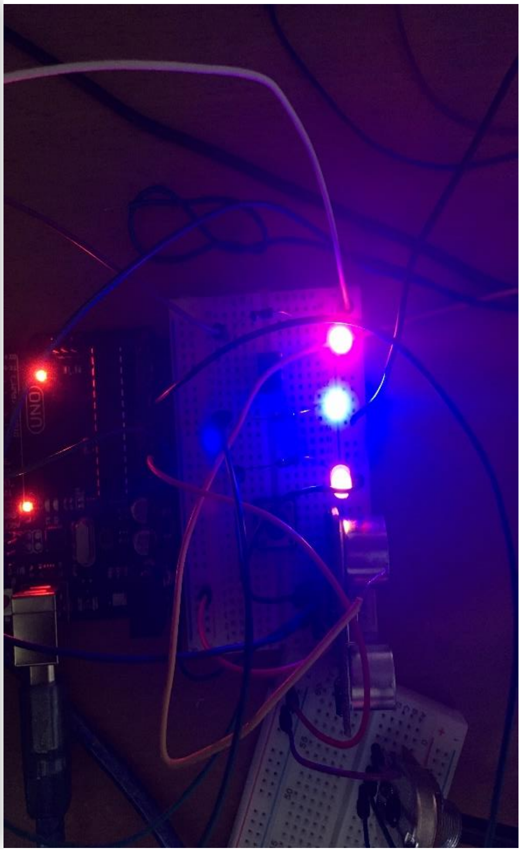
- Code Processing :



```
import controlP5.*;
import processing.serial.*;

Serial port;

ControlP5 cp5;// creating a controlP5 object
PFont font;

void setup(){
  size(300,500);

  printArray(Serial.list());//prints out all available ports

  port=new Serial(this,"COM5",9600);

  cp5=new ControlP5(this);
  font = createFont("calibri light",20);
  cp5.addButton("red") ///the name of he button  text "red"
      .setPosition(60,50)// x,y coordinates
      .setSize(180,80)// the width and height
      .setFont(font)// adding a font size
      ;
  cp5.addButton("yellow").setPosition(60,150).setSize(180,80).setFont(font);
    cp5.addButton("blue").setPosition(60,250).setSize(180,80).setFont(font);
      cp5.addButton("alloff").setPosition(60,350).setSize(180,80).setFont(font);
}
void draw(){
  background(150,0,150);//bg color

  // title
  fill(255,255,255);// rgb color for the text
  text("LED CONTROL",100,30); // (text,x,y)

}

// functions that relates to every button declared
void red(){port.write('r');}
void blue(){port.write('b');}
void yellow(){port.write('y');}
void alloff(){port.write('f');}
```

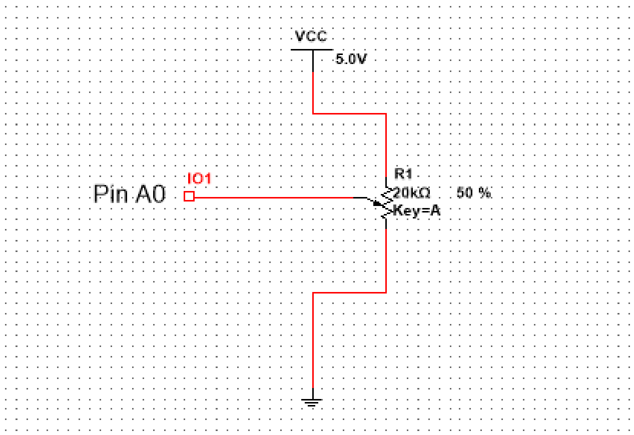
- Résultat :



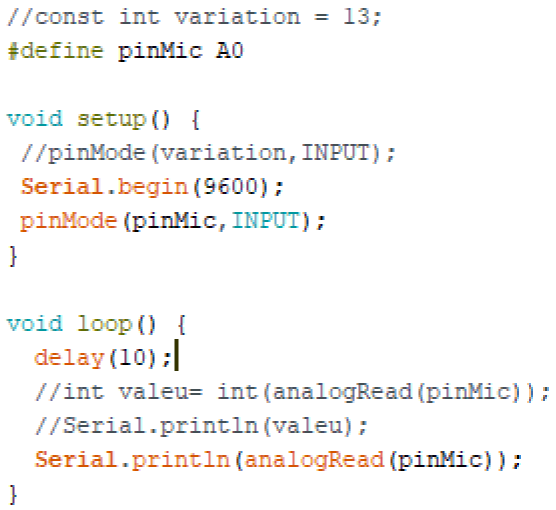


2. Dessiner avec un potentiomètre avec processing

- Branchement :

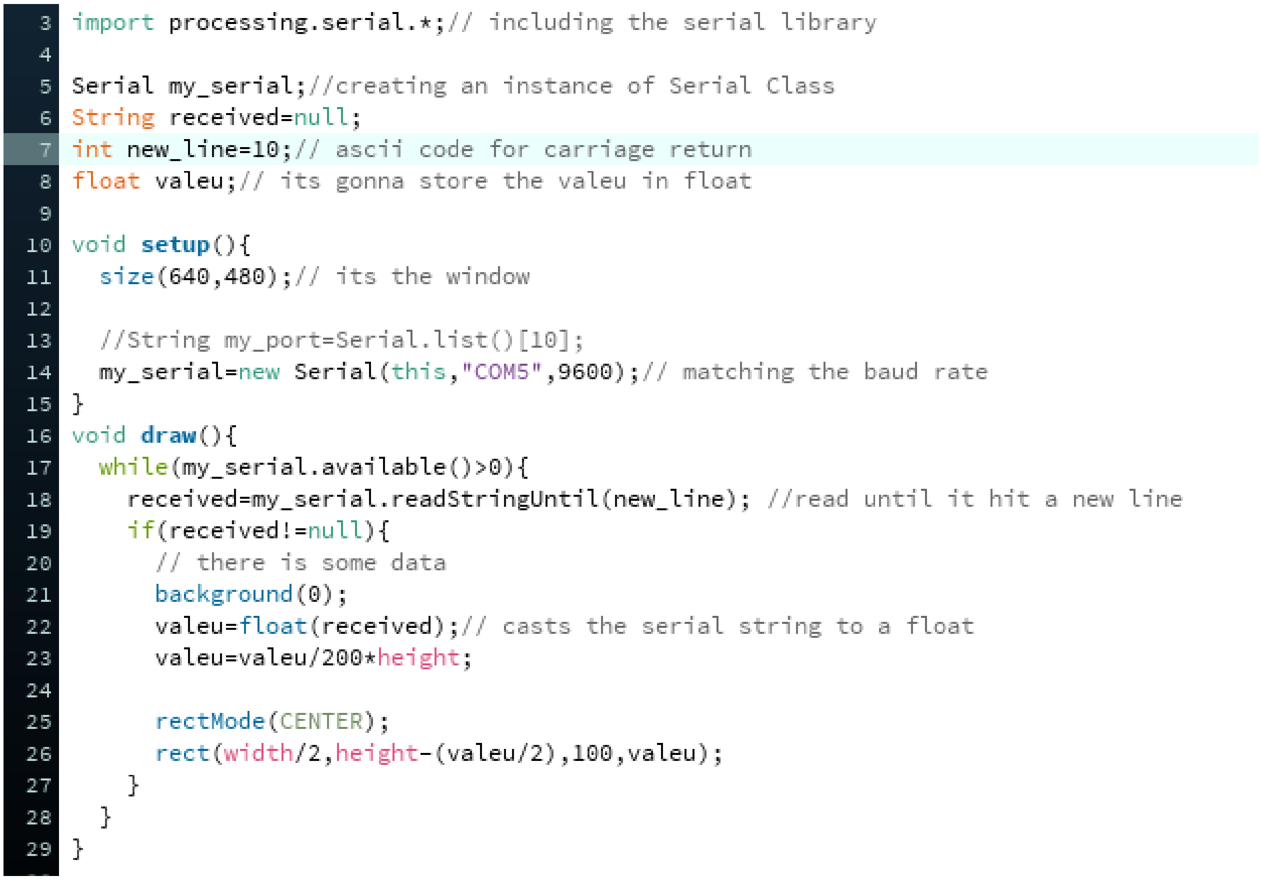


- Code Arduino :



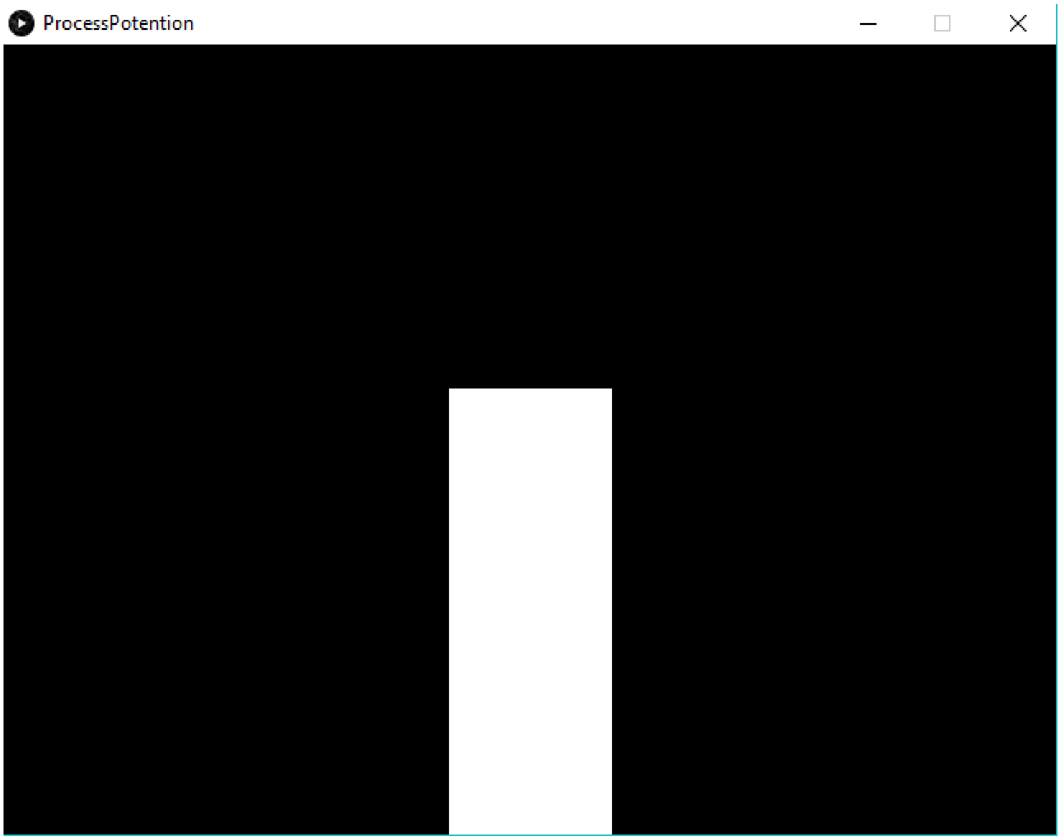
```
//const int variation = 13;
#define pinMic A0

void setup() {
 //pinMode(variation,INPUT);
 Serial.begin(9600);
 pinMode(pinMic,INPUT);
}

void loop() {
  delay(10);
  //int valeu= int(analogRead(pinMic));
  //Serial.println(valeu);
  Serial.println(analogRead(pinMic));
}
```

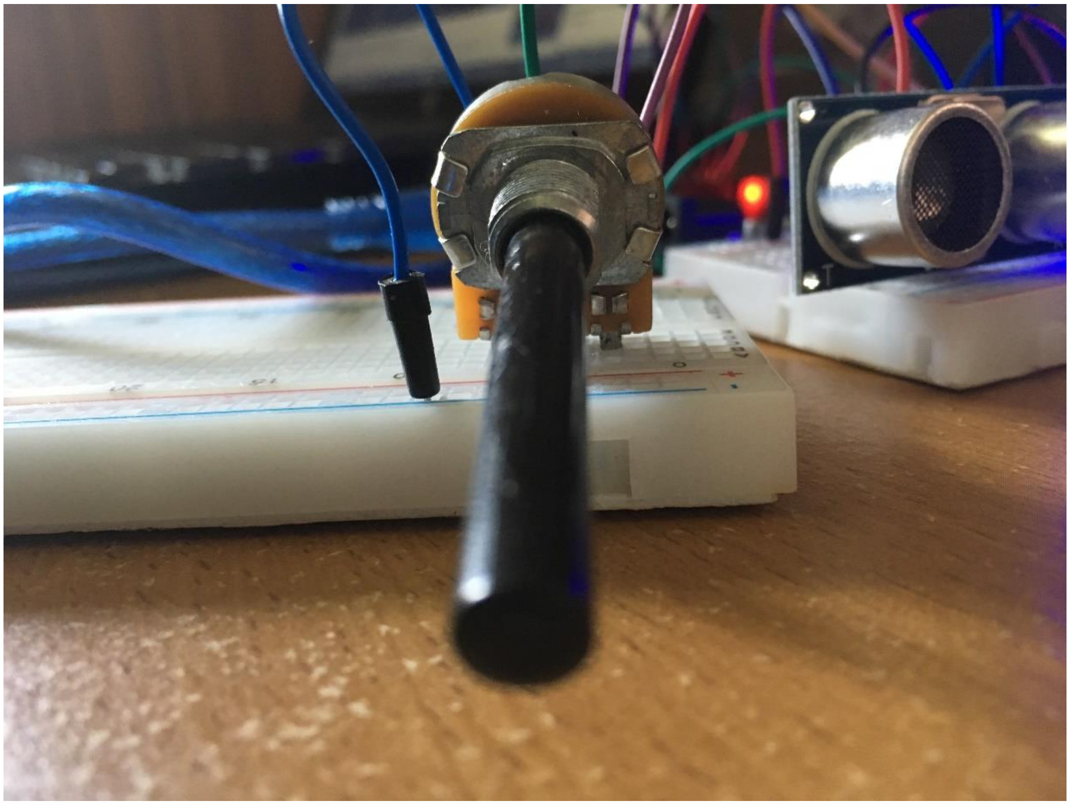
- Code Processing :



```
import processing.serial.*;// including the serial library

Serial my_serial;//creating an instance of Serial Class
String received=null;
int new_line=10;// ascii code for carriage return
float valeu;// its gonna store the valeu in float

void setup(){
  size(640,480);// its the window

  //String my_port=Serial.list()[10];
  my_serial=new Serial(this,"COM5",9600);// matching the baud rate
}
void draw(){
  while(my_serial.available()>0){
    received=my_serial.readStringUntil(new_line); //read until it hit a new line
    if(received!=null){
      // there is some data
      background(0);
      valeu=float(received);// casts the serial string to a float
      valeu=valeu/200*height;

      rectMode(CENTER);
      rect(width/2,height-(valeu/2),100,valeu);
    }
  }
}
```

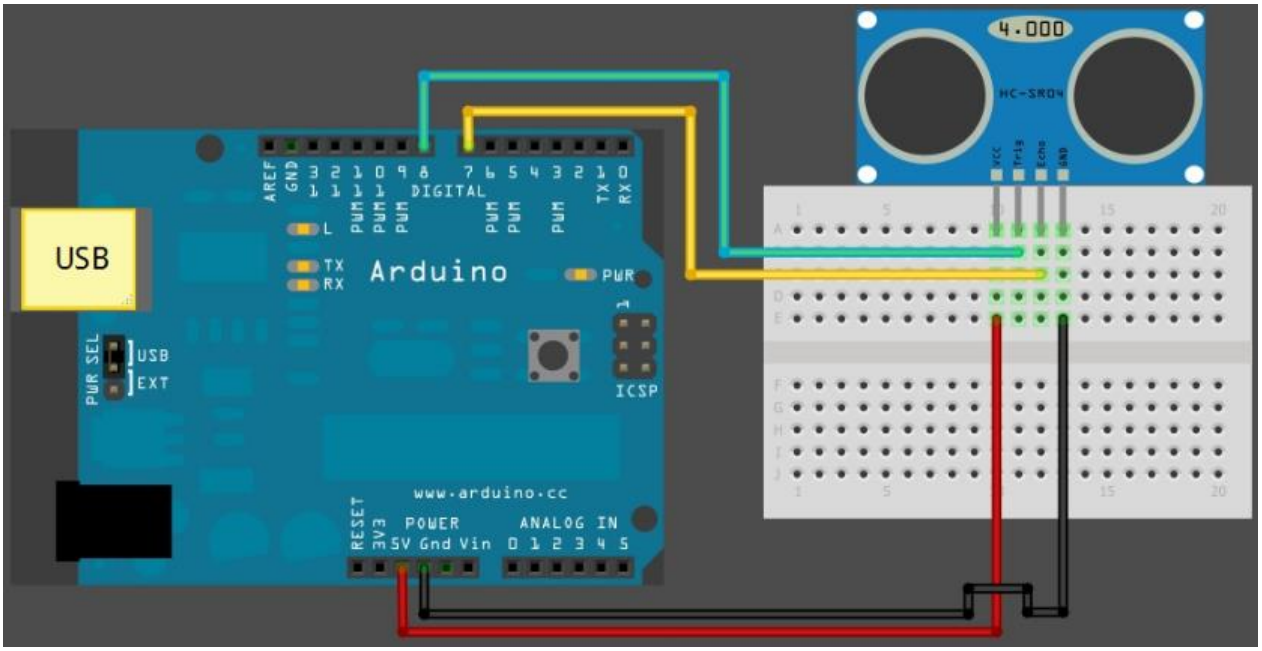
- Résultat :



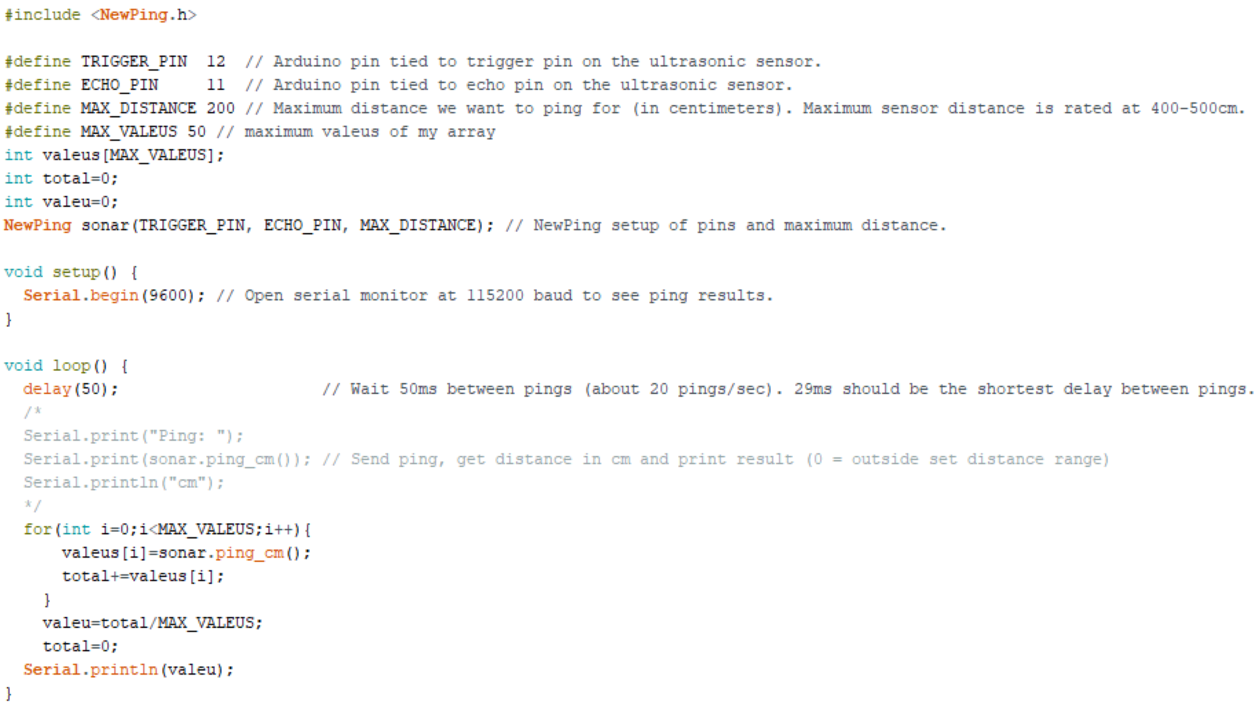


3. Détecter des objets à une distance avec Processing

- Utilisation du capteur ultra-sons :

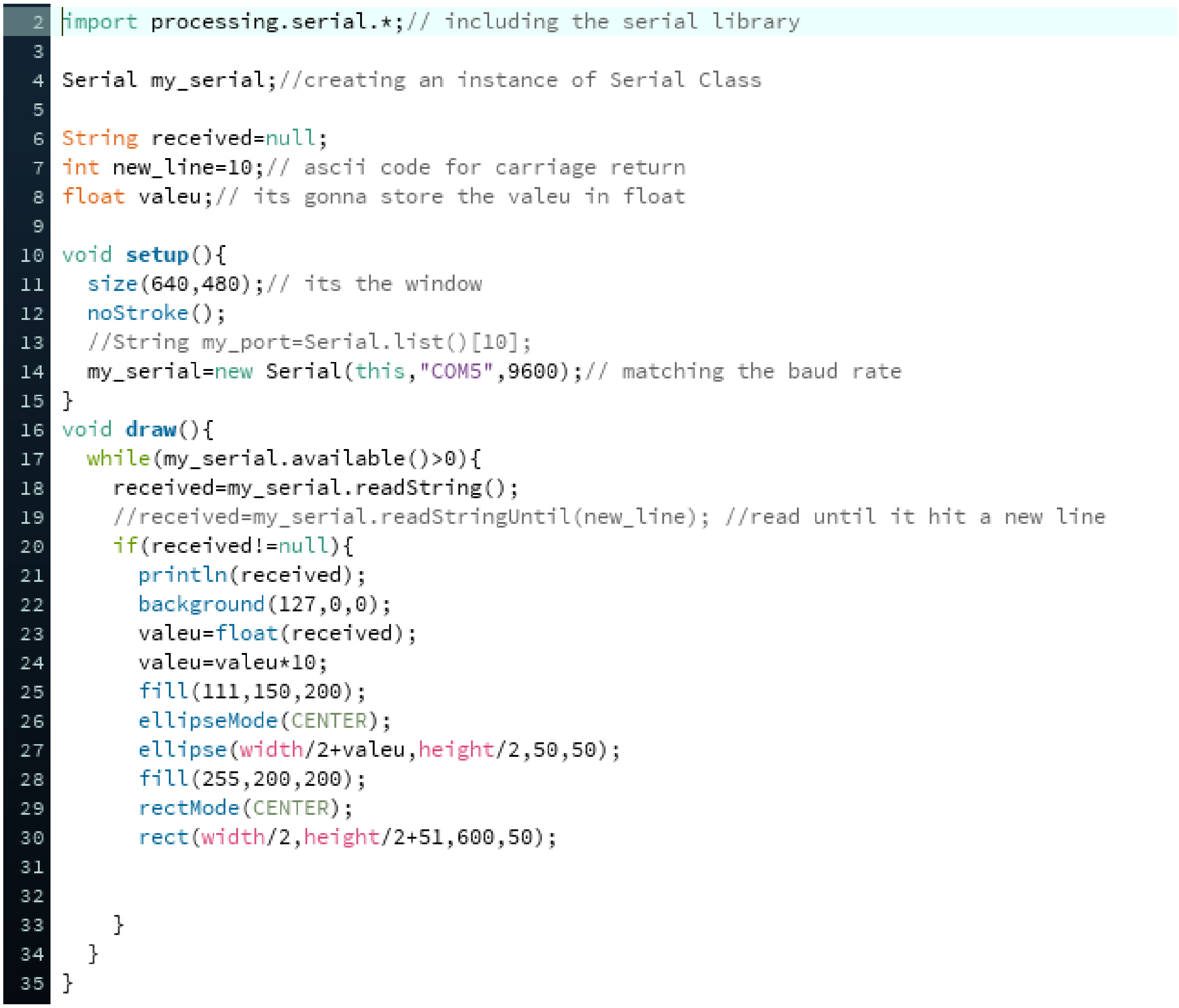


- Code Arduino :



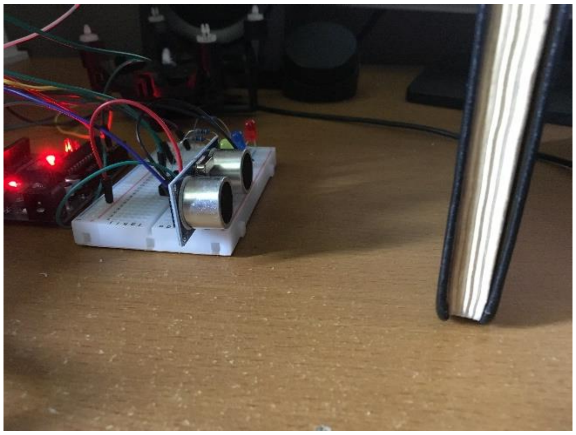
```
#include <NewPing.h>

#define TRIGGER_PIN  12  // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN     11  // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at 400-500cm.
#define MAX_VALEUS 50 // maximum valeus of my array
int valeus[MAX_VALEUS];
int total=0;
int valeu=0;
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.

void setup() {
  Serial.begin(9600); // Open serial monitor at 115200 baud to see ping results.
}

void loop() {
  delay(50);                      // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay between pings.
  /*
  Serial.print("Ping: ");
  Serial.print(sonar.ping_cm()); // Send ping, get distance in cm and print result (0 = outside set distance range)
  Serial.println("cm");
  */
  for(int i=0;i<MAX_VALEUS;i++){
      valeus[i]=sonar.ping_cm();
      total+=valeus[i];
    }
    valeu=total/MAX_VALEUS;
    total=0;
  Serial.println(valeu);
}
```

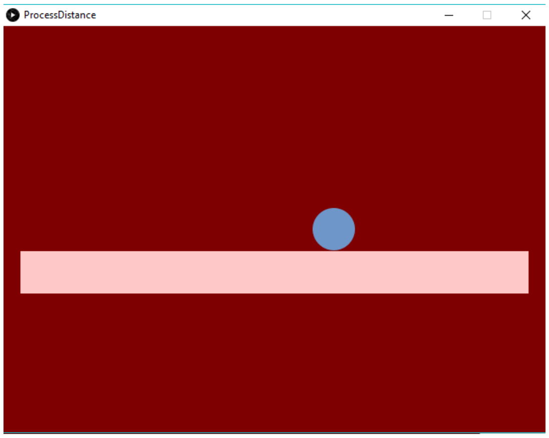
- Code Processing :



```
import processing.serial.*;// including the serial library

Serial my_serial;//creating an instance of Serial Class

String received=null;
int new_line=10;// ascii code for carriage return
float valeu;// its gonna store the valeu in float

void setup(){
  size(640,480);// its the window
  noStroke();
  //String my_port=Serial.list()[10];
  my_serial=new Serial(this,"COM5",9600);// matching the baud rate
}
void draw(){
  while(my_serial.available()>0){
    received=my_serial.readString();
    //received=my_serial.readStringUntil(new_line); //read until it hit a new line
    if(received!=null){
      println(received);
      background(127,0,0);
      valeu=float(received);
      valeu=valeu*10;
      fill(111,150,200);
      ellipseMode(CENTER);
      ellipse(width/2+valeu,height/2,50,50);
      fill(255,200,200);
      rectMode(CENTER);
      rect(width/2,height/2+51,600,50);


    }
  }
}
```

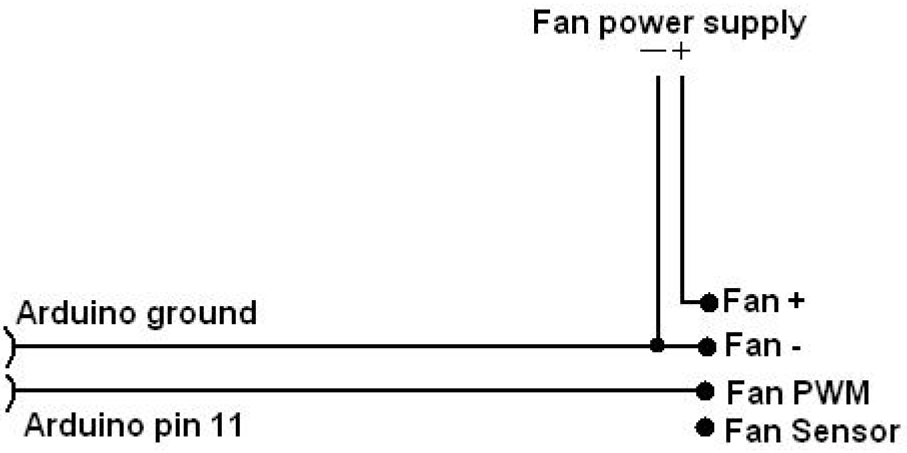
- Résultat :





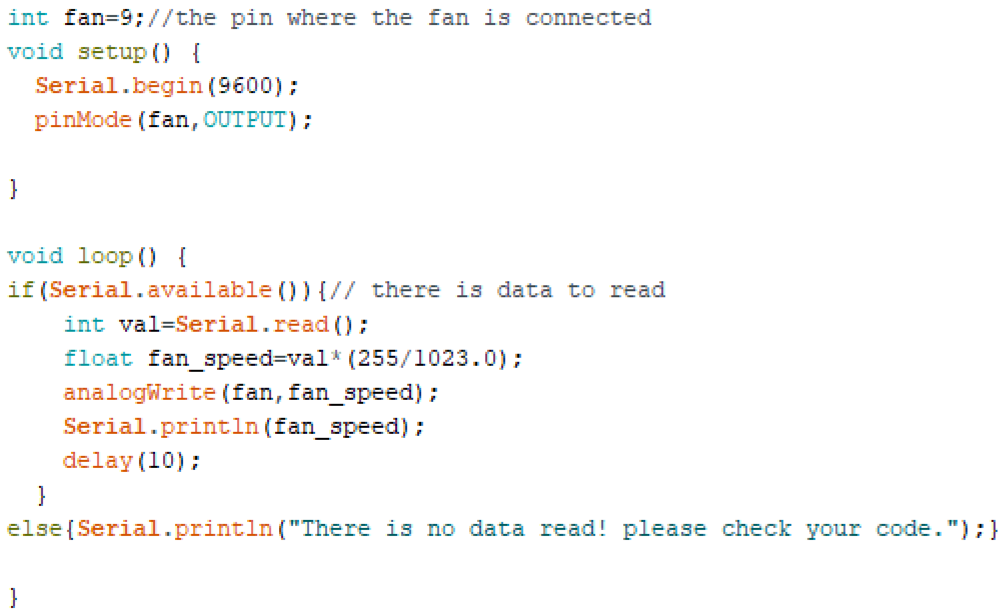
4. Contrôler un ventilateur avec Processing

- Branchement : (J'ai récupéré un fan de mon vielle tour)



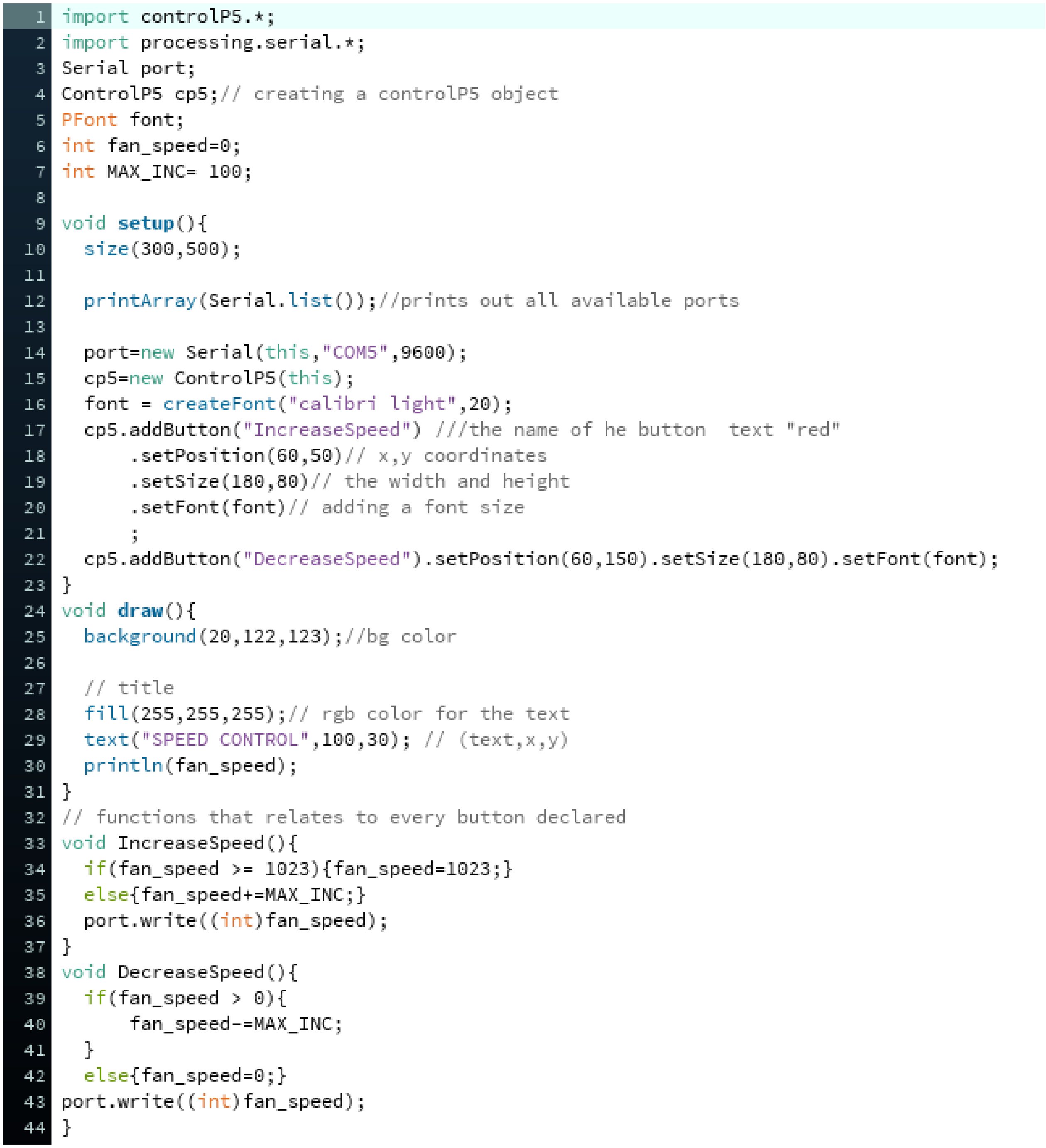
(Je utiliser la Pin 9 sur Arduino, juste parce que les autres étés déjà utiliser)

- Code Arduino :



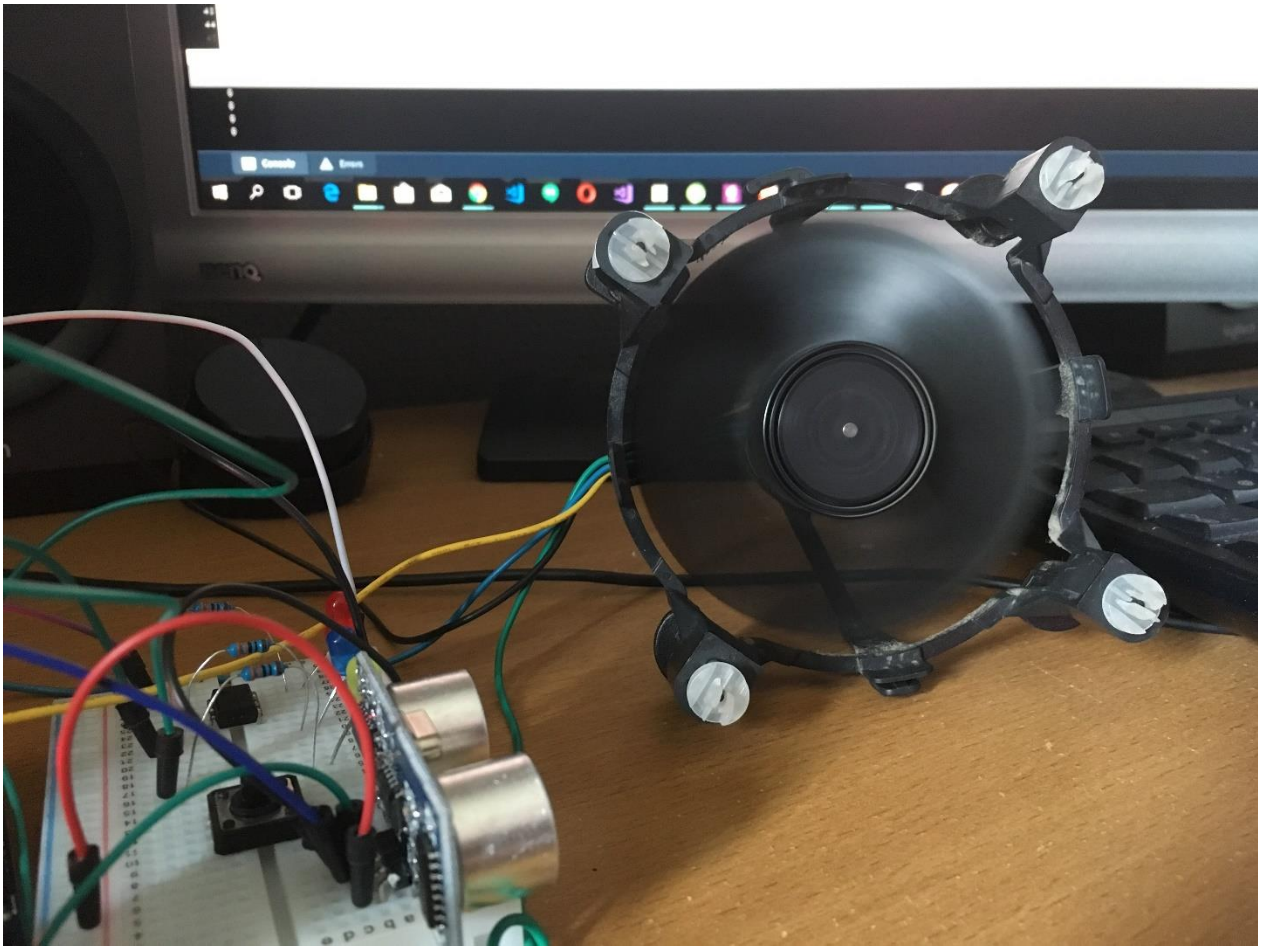
```
int fan=9;//the pin where the fan is connected
void setup() {
  Serial.begin(9600);
  pinMode(fan,OUTPUT);

}

void loop() {
if(Serial.available()){// there is data to read
    int val=Serial.read();
    float fan_speed=val*(255/1023.0);
    analogWrite(fan,fan_speed);
    Serial.println(fan_speed);
    delay(10);
  }
else{Serial.println("There is no data read! please check your code.");}

}
```

- Code Processing :



```
import controlP5.*;
import processing.serial.*;
Serial port;
ControlP5 cp5;// creating a controlP5 object
PFont font;
int fan_speed=0;
int MAX_INC= 100;

void setup(){
  size(300,500);

  printArray(Serial.list());//prints out all available ports

  port=new Serial(this,"COM5",9600);
  cp5=new ControlP5(this);
  font = createFont("calibri light",20);
  cp5.addButton("IncreaseSpeed") ///the name of he button  text "red"
      .setPosition(60,50)// x,y coordinates
      .setSize(180,80)// the width and height
      .setFont(font)// adding a font size
      ;
  cp5.addButton("DecreaseSpeed").setPosition(60,150).setSize(180,80).setFont(font);
}
void draw(){
  background(20,122,123);//bg color

  // title
  fill(255,255,255);// rgb color for the text
  text("SPEED CONTROL",100,30); // (text,x,y)
  println(fan_speed);
}
// functions that relates to every button declared
void IncreaseSpeed(){
  if(fan_speed >= 1023){fan_speed=1023;}
  else{fan_speed+=MAX_INC;}
  port.write((int)fan_speed);
}
void DecreaseSpeed(){
  if(fan_speed > 0){
      fan_speed-=MAX_INC;
  }
  else{fan_speed=0;}
port.write((int)fan_speed);
}
```

Résultat :



Conclusion :

On peut conclure en disant que on peut utiliser l’Arduino aussi pour visualiser et analyser des données.

L’Art et aussi possible avec l’Arduino + Prossessing (P5.js).