Lab Note Book

Poworm Rangers

Biosensors - Project Forces

Nina Guérin
Clément Conil
Samuel Churlaud

Start of the second project of Biosensors on Tuesday 24th of January 2017. End on wednesday 1st of February 2017.

The aim of the project is to compare a biological sensor with a electronic sensor. The theme of the project is Forces. It includes sound, vibration, touch, water flow, etc.

This lab-notebook is intended to give an overview of the experiment we led, what happened and what problems we encountered. It has purpose to make this project replicable by anyone.

All the code and definitive protocol are available in the github project: here[1].

PoWorm Rangers

The team PoWorm Rangers is composed by 3 bachelor FDV student: Nina Guérin, Clément Conil and Samuel Churlaud.

Our project aim is to determine which, between a piezoelectric element and and *Chironomus plumosus* larvae, is the better to detect intensity and direction of vibrations.

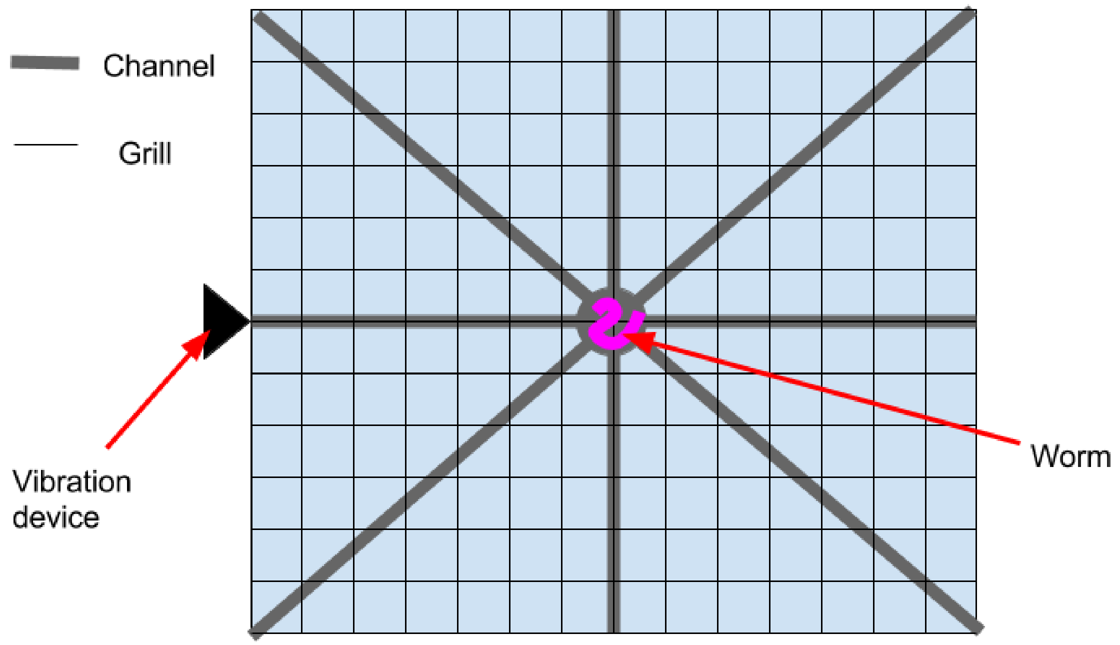
[1] http://github.com/learningthruresearch/Biosensors2017/tree/master/poworm_rangers

Tuesday 24 of January

First team meeting. We decided to work on *Lumbriculus variegatus (*aka bloodworm) and its sensitivity to vibrations. Indeed, we learnt that bloodworm can detect vibration and goes out of the ground when there is vibrations. This is a method often used in order to make worms leave.[2]

Our first protocol is described below:

We will use a square plan area (named observation area), divided in several subsquares and with 6 channels representing the directions where the worm can go.



We put a *Lumbriculus variegatus* in the middle of the grid. We put a motor device that makes vibration when controlled by an Arduino. The worm will have the choice of 6 different directions to flee or go towards the origin of the vibration. Then, we record the movement with a camera and analyse it with a tracking software : speed of movement and distance between the vibration device and the worm.

Tested intensity: we will test 3 different intensities : low, medium and high (unities to define)

---

[2] 'Des Vibrations Qui Font Fuir Le Ver Hors de Terre'. *Sciences et Avenir*. Accessed 27 January 2017. http://www.sciencesetavenir.fr/nature-environnement/des-vibrations-qui-font-fuir-le-ver-hors-de-terre_4323.

Positive control : we put the *Lumbriculus variegatus* in the middle and switch on the vibrating device at maximum intensity (intensity = amplitude in décibel). We film the *Lumbriculus variegatus* during 3 minutes and observe its moving.

Negative control : we put the *Lumbriculus variegatus* in the center and film during 3 minutes without activating the vibrating device.

Step of the experiment for biological sensor:

1. Put the worm
2. Start the camera
3. Activate vibrating device at intensity 1 during 3 minutes
4. Repeat step 1 to 3 five times in order to have 5 replicates, changing the worm each time
5. Change worm
6. Repeat step 1 to 4 with intensity 2
7. Change worm
8. Repeat step 1 to 4 with intensity 3
9. Set the worm free

Then, we repeat the experiment with the Piezo vibration sensor instead of the worm. We put the piezo in the middle of the grid. With a voltmeter, we gather the value given by the piezo and then have an intensity of vibration.

We will compare the value given by the piezo and the movement of the worm to the value of vibration that we put on the arduino controlling the vibrating device.

<u>Wednesday 25 of January</u>

We realized that we couldn't obtain bloodworm at this season of the year, so we decided to work on *Chironomus plumosus* larvae. Those worm are usually used to feed fishes, this is why we can buy it in pet shops. One of our comrade, Elena, went to the Truffaut at Quai de la Gare to buy 2 pack of 90 mL of worms for us.

In order to preserve the worm before experiment, we put them in a jar after filtrated dead worms. We let the jar open with some additional tape water.

We also prepared the definitive protocol for biological and electronical sensor. Those protocols are available in the project's github.

Friday 27th of January

We started preparing devices for the experiment.

Electronic sensor: piezo element.

In order to build piezo sensor, we used the following part[3]:

- 1 piezoelectric element
- 4 electrical wire
- 1 breadboard
- 1 Arduino
- 1 M resistor

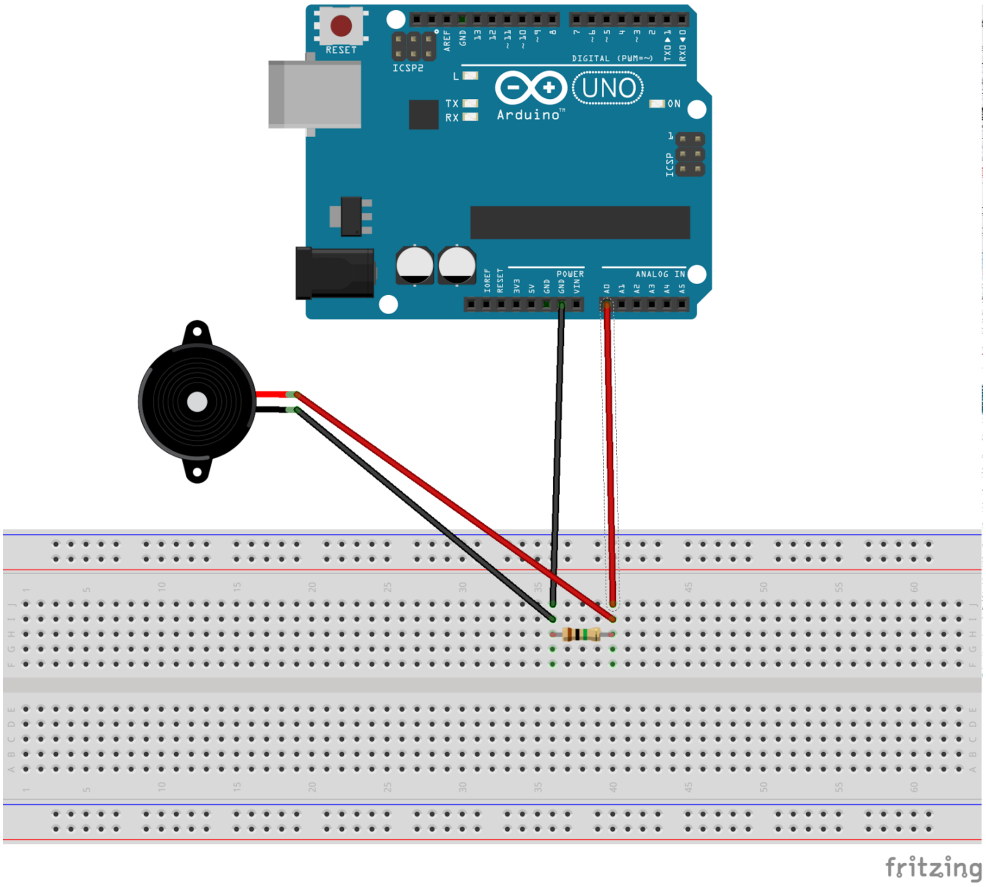
For the piezo sensor, we used a simple setup like shown in the schema below.

The piezo element has a white sensor in ceramic at its center surrounded by a golden circle as a support.

We put the ground wire on the support and the sensor on the analog 0 port. The resistance is here to protect the sensor and the Arduino to a too strong electric courant.

Wire	Sensor port	Arduino Port
Black	Support (edge circle)	Ground
Resistance 1M	-	Ground and Analog 0
Red	Sensor	Analog 0

[3] 'Arduino - Knock'. Accessed 26 January 2017. https://www.arduino.cc/en/Tutorial/Knock.



We will put the sensor in the middle of the aluminium plate and make the plate vibrate with the vibration motor. We will then record the intensity of vibration thanks to the piezo programmed with the following arduino code :

```
/******************************************************************************
Piezo_Vibration_Sensor.ino
Example sketch for SparkFun's Piezo Vibration Sensor
  (https://www.sparkfun.com/products/9197)
Jim Lindblom @ SparkFun Electronics
April 29, 2016

- Connect a 1Mohm resistor across the Piezo sensor's pins.
- Connect one leg of the Piezo to GND
- Connect the other leg of the piezo to A0

Vibrations on the Piezo sensor create voltags, which are sensed by the Arduino's
A0 pin. Check the serial monitor to view the voltage generated.

Development environment specifics:
Arduino 1.6.7
******************************************************************************/
const int PIEZO_PIN = A0; // Piezo output

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  // Read Piezo ADC value in, and convert it to a voltage
  int piezoADC = analogRead(PIEZO_PIN);
  float piezoV = piezoADC / 1023.0 * 5.0;
  Serial.println(piezoV); // Print the voltage.
}
```

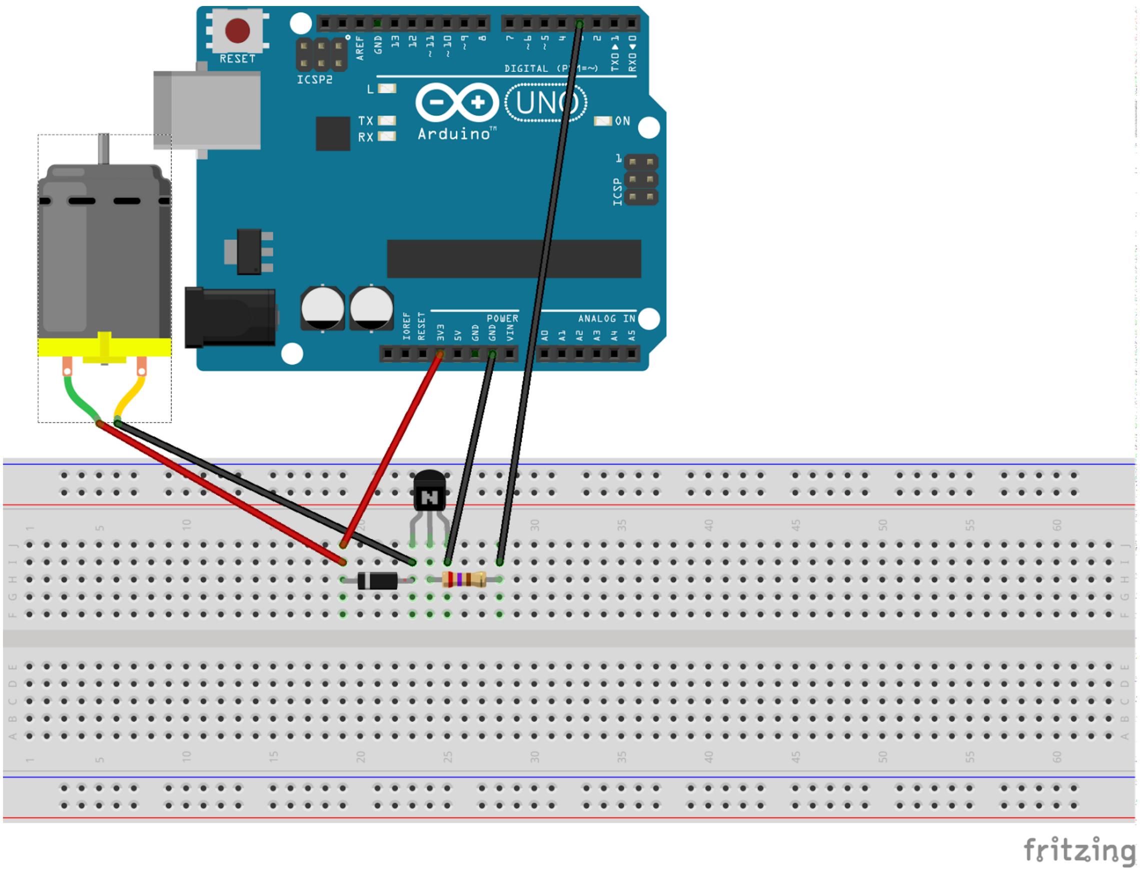
By opening the serial monitor, you can read and collect datas of voltage provoque by vibrations.

<u>Vibration generator:</u>

In order to generate vibration, we use the guides lines on adafruit.com[4]. The tutorial permit to build a system with arduino and 6V DC motor. We used a vibrator motor instead.
The building of the vibrator generator requires the following part:

- One vibration motor
- One 2N2222 transistor
- One 1N4001 diode
- One 270 Ω resistor
- One breadboard
- 3 wires

We built the following setup:



Warning: the motor used for the experiment is not a rotating motor as you can see in the schemes but a vibrating motor.

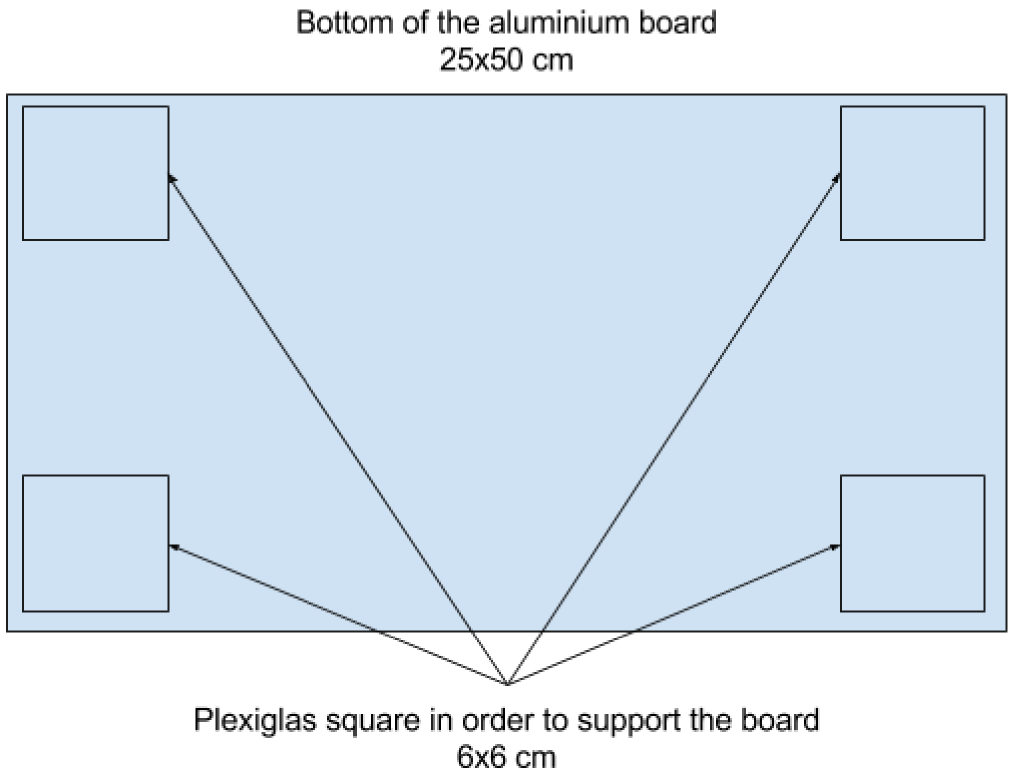
We upload on the arduino the following code[5]:

[4] 'Overview | Arduino Lesson 13. DC Motors | Adafruit Learning System'. Accessed 27 January 2017. https://learn.adafruit.com/adafruit-arduino-lesson-13-dc-motors/overview.
[5] 'Piezo Vibration Sensor Hookup Guide - Learn.sparkfun.com'. Accessed 26 January 2017. https://learn.sparkfun.com/tutorials/piezo-vibration-sensor-hookup-guide/example-code.

code-moteur §

```
/*
Adafruit Arduino - Lesson 13. DC Motor
*/


int motorPin = 3;

void setup()
{
  pinMode(motorPin, OUTPUT);
  Serial.begin(9600);
  while (! Serial);
  Serial.println("Speed 0 to 255");
}


void loop()
{
  if (Serial.available())
  {
    int speed = Serial.parseInt();
    if (speed >= 0 && speed <= 255)
    {
      analogWrite(motorPin, speed);
    }
  }
}
```

<u>Build of the channel device:</u>
We buy an aluminium board, which had a size of 25x50 cm. We add below it four square pieces of plexiglas in order to avoid contact between table and aluminum board.



We try to stick the plexiglas foot with glue, but it didn't work. So we just put the plexiglas piece under the board, without fixing it.

We tried to prepare a first data acquisition of electronical sensor data, but we encountered several problems.

- The two leonardo in which our piezo element were plug were not working, so we had to changed it with new arduinos.
- If we plug two arduino to only one computer, we cannot access to the monitor of the two arduino at the same time. So in order to gather data from the 2 arduino and changing vibration intensity, we need 3 computers.
- We were at the end of the day and we lacked of time.

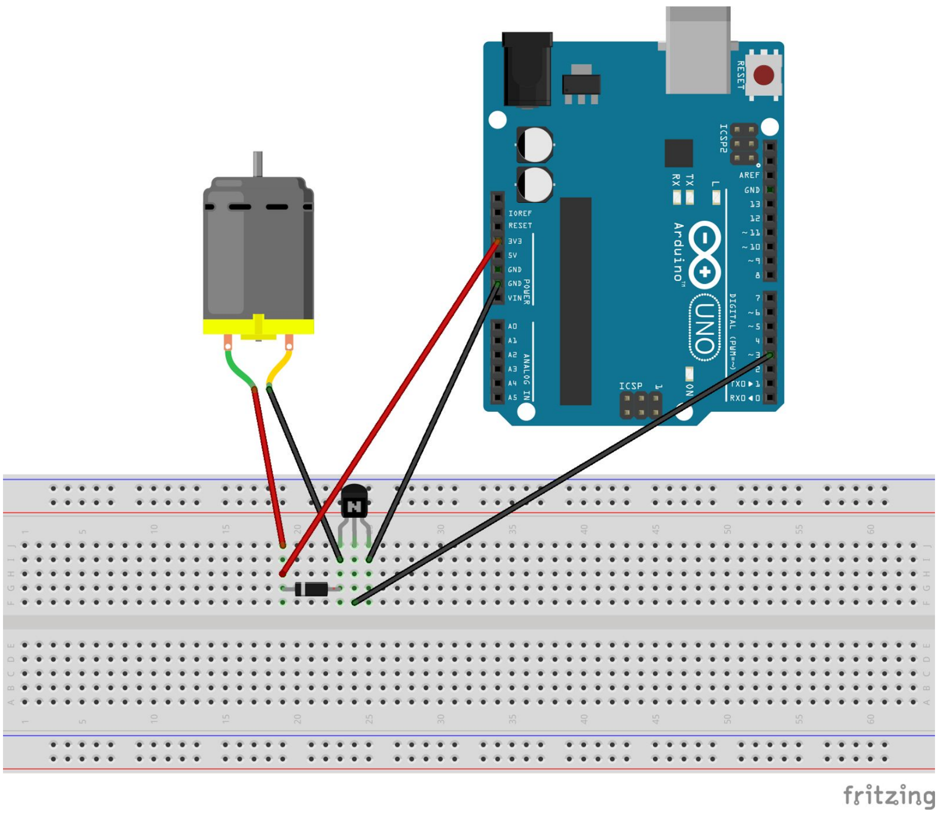
Because of this several reasons, we had to do data acquisition on Saturday.

Saturday 28th of January

We proceed to the data acquisition but we encountered several problems.

First, we had to change the motor. Indeed, instead of a 6V motor as shown in the scheme we used to build our motor system, but the vibrator motor we needed was 3.3V motor, so we had to do several change: we took back the resistor and we modified the code arduino in order to provoque vibration during precisely 4 second and then stop.

Here the new circuit:



And here the new code, available on the github and named **code-moteur.ino**[6]:

[6] Available here:
https://github.com/learningthruresearch/Biosensors2017/tree/master/poworm_rangers/code/code-moteur.ino

```
/*
Adafruit Arduino - Lesson 13. DC Motor
*/


int motorPin = 3;
int speed = 0 ;

void setup()
{
  pinMode(motorPin, OUTPUT);
  Serial.begin(9600);
  while (! Serial);
  Serial.println("Speed 0 to 255");
}


void loop()
{
  if (Serial.available())
  {
    speed = Serial.parseInt();
    if (speed >= 0 && speed <= 255)
    {
      analogWrite(motorPin, speed);
      delay(4000);
    }
  }
}
```

Second, we encounter also problem with the arduino code of the piezo element. We couldn't modified the sensibility threshold of the piezo. So we decided to merge two code we found online: the one we used precedently, and the code for the knock sensor[7].

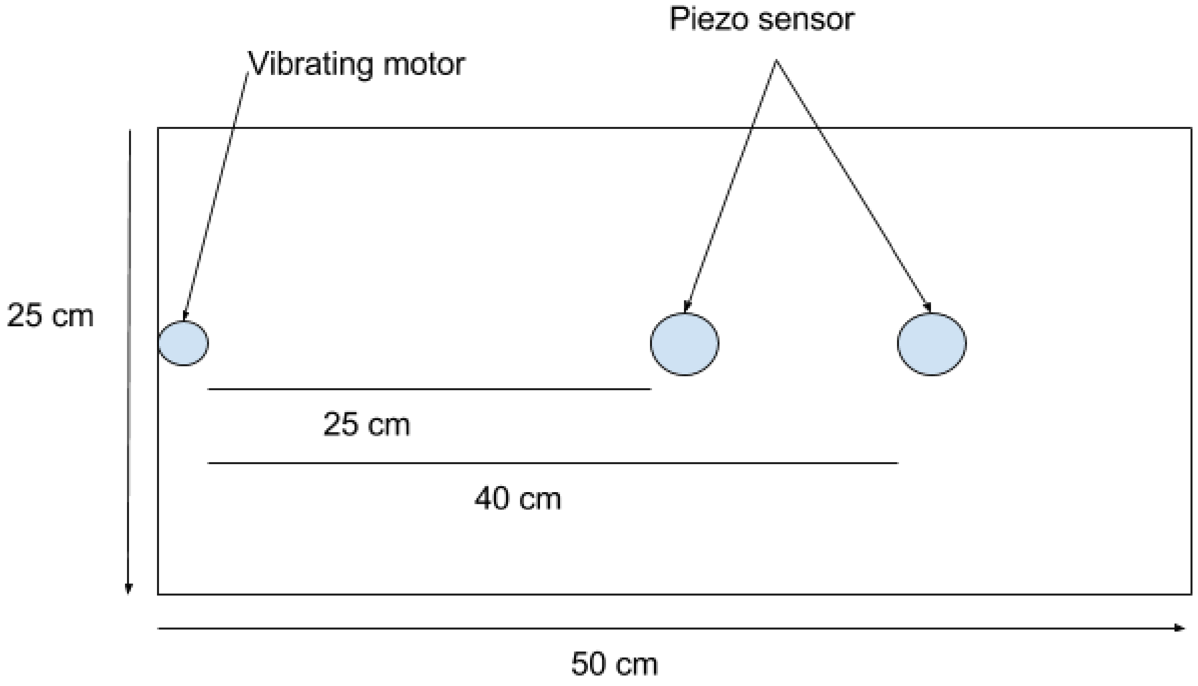
Here the new code, named **code-piezo.ino**[8].,

```
/*
 *  Piezo Sensor
 */


// these constants won't change:
const int knockSensor = A0; // the piezo is connected to analog pin 0
const int threshold = 1;  // threshold value to decide when the detected sound is a knock or not


// these variables will change:
int sensorReading = 0;      // variable to store the value read from the sensor pin

void setup() {
  Serial.begin(9600);       // use the serial port
}

void loop() {
  // read the sensor and store it in the variable sensorReading:
  sensorReading = analogRead(knockSensor);

  // if the sensor reading is greater than the threshold:
  //if (sensorReading >= threshold) {
    // send the string "Knock!" back to the computer, followed by newline
    //Serial.println("Knock!");
  //}
  Serial.println(sensorReading);
  delay(100);  // delay to avoid overloading the serial port buffer
}
```

[7] 'Arduino - Knock'. Accessed 26 January 2017. https://www.arduino.cc/en/Tutorial/Knock.

[8] Available here: https://github.com/learningthruresearch/Biosensors2017/tree/master/poworm_rangers/code/code-piezo.ino

Once those problems were solved, we could proceed to the data acquisitions for the electrical sensor.
We placed the vibrator motor at the extremity of the aluminium board.
We placed one piezo at 25 cm to the vibrating motor (in the center of the board), and the second one to 40 cm.
You can see the set-up in the schemes below:



To acquire data, we used a python code, that read and copy data received in the arduino monitor in a csv file every 0.1 seconds.
Here the python code we used, available in the github and named **electronic_collecting.py** [9]:

```
# coding = UTF-8
# written by Samuel Churlaud
# see github.com/learningthruresearch/Biosensors2017/tree/master/poworm_rangers

import serial
import time

ser = serial.Serial('/dev/ttyACM0', 9600)
output = open('25_255_3', 'wb')#distance_v moteur_réplicat
for i in range(100):#à peu près 10 secondes
    data = ser.readline()
    output.write(data)
    time.sleep(0.1)
```

[9] Available here: https://github.com/learningthruresearch/Biosensors2017/tree/master/poworm_rangers/code/electronic_collecting.py

Thanks to the teams Terra Force and Carnisensors for sharing their python codes with us, it was very helpful.

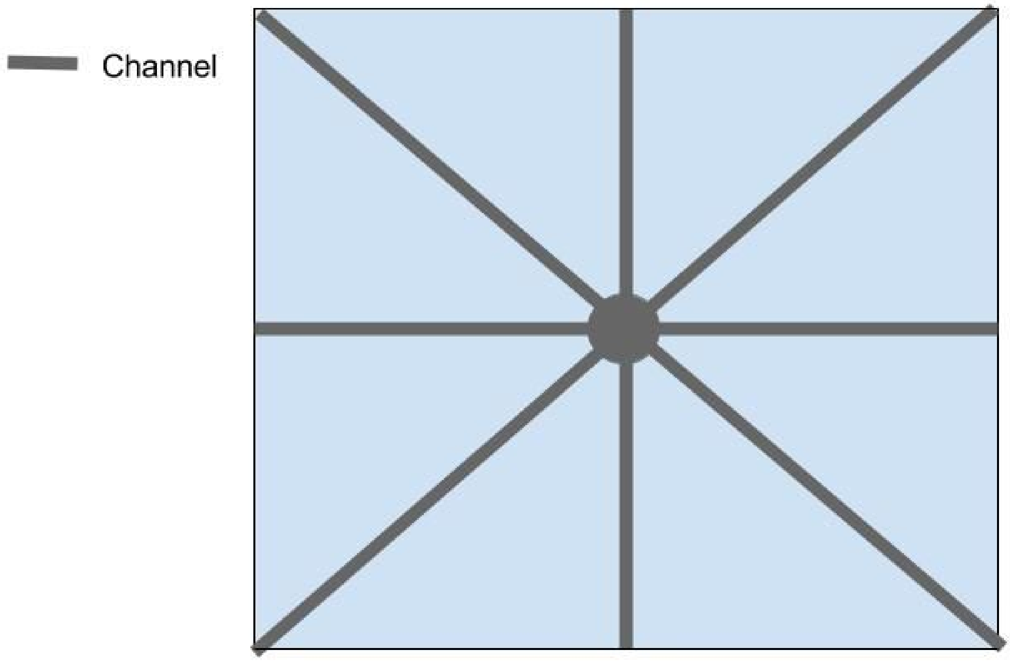
We tested 5 intensity of vibration: 0 (negative control), 100, 150, 200 and 255 (positive control). For each intensity we did 3 replicates.

Protocol:
In the python code, name the file that will contain the data.
Start the data acquisition by launching the python code.
Wait 3 seconds and then launch the wished vibration intensity value in the motor’s arduino monitor.
Wait until the python code stop.

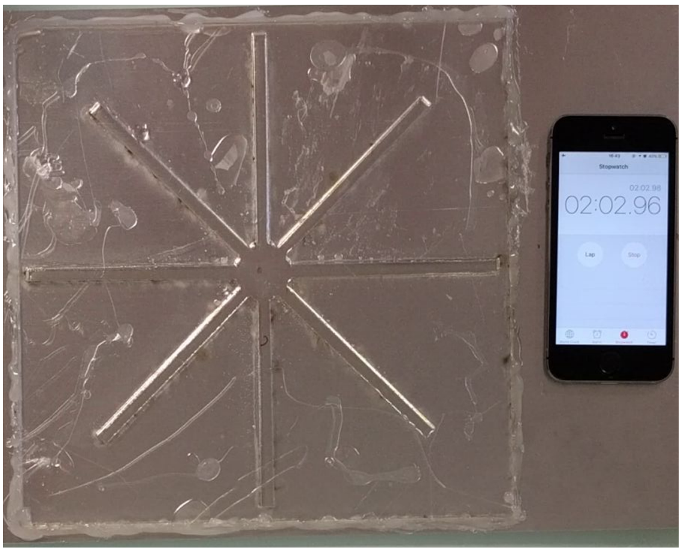
Monday 30th of January

During this day, we prepared the material in order to test the biological sensor.

First we had to build the device. For our experiment, we wanted to measure the distance between the worm and the sources of the vibration. We needed a system of channels in order to control the direction in which the worm could go.
With the laser cutter, we cut in a plexiglas board the channel system downloadable on the github[10]. It look like this:



After this, with glue the channels on the aluminium board, the center at 25 cm of the motor.



[10] https://github.com/learningthruresearch/Biosensors2017/blob/master/poworm_rangers/plaquepoworm.cdr

We had to start late because we had the sadness to discover that the worm we bought few days ago were dead. So we had to buy some again at Truffaut.

Finally we succeeded to obtain a first set of biological data. To do that, we started shooting with a camera the channels device, in center of which we had put the worm. Acquisition was during two minutes, started when we put the worm. The vibration started after 15 seconds of recording. Worms were not harm during the experiment.

Tuesday 31st January

We collected the datas in the lab. Thanks to the previous work we did on Monday, we only had to do four (and more if possible) replicates to do by intensity of vibration.

Using the same device and protocol than on Monday, we obtained in total 5 replicates for each intensity of vibrations.

We had to analyze the video. In order to do so, we sliced each video in a certain number of images.

We then analysed these images on ImageJ. We tracked down the position of the worm and measured the distance between it and the beginning of the plexiglass device in the side of the motor. We gathered this distance in 29 csv files.

Then, we plotted these datas as a distance between the worm and the vibration motor as a function of time with the following python code **bio_analysis.py**[11]:

```
# coding = UTF-8
# written by Samuel Churlaud
# see github.com/learningthruresearch/Biosensors2017/tree/master/poworm_rangers
# version 1.0

# modules
import csv
import matplotlib.pyplot as plt
import matplotlib.lines as mlines

# variables
listVibrations = [0, 100, 150, 200, 255]
replicates = ['A', 'B', 'C', 'D', 'E', 'F']
average0 = []
average100 = []
average150 = []
average200 = []
average255 = []
time = []


# functions
def results0():
        for i in range(29):
        average0.append(int(0))

        for i in replicates:
```

[11] Available here: https://github.com/learningthruresearch/Biosensors2017/blob/master/poworm_rangers/code/bio_analysis.py

```
        reader = csv.reader(open('clean_bio_sensor/Results0' + i + '.csv', 'r'))
        for line in reader:
        print(line[0])
        if int(line[0]) <= 29:
                average0[int(line[0]) - 1] += float(line[1])
        else:
                pass


def results100():

        for i in range(29):
        average100.append(int(0))

        for i in replicates:
        reader = csv.reader(open('clean_bio_sensor/Results100' + i + '.csv', 'r'))
        for line in reader:
        print(line[0])
        if int(line[0]) <= 29:
                average100[int(line[0]) - 1] += float(line[1])
        else:
                pass


def results150():
        for i in range(29):
        average150.append(int(0))

        for i in replicates:
        reader = csv.reader(open('clean_bio_sensor/Results150' + i + '.csv', 'r'))
        for line in reader:
        print(line[0])
        if int(line[0]) <= 29:
                average150[int(line[0]) - 1] += float(line[1])
        else:
                pass


def results200():
        for i in range(29):
        average200.append(int(0))

        for i in replicates:
        if i == 'B':
        pass
        else:
        reader = csv.reader(open('clean_bio_sensor/Results200' + i + '.csv', 'r'))
        for line in reader:
                print(line[0])
                if int(line[0]) <= 29:
```

```
                    average200[int(line[0]) - 1] += float(line[1])
                    else:
                    pass


def results255():
          for i in range(29):
          average255.append(int(0))

          for i in replicates:
          reader = csv.reader(open('clean_bio_sensor/Results255' + i + '.csv', 'r'))
          for line in reader:
          print(line[0])
          if int(line[0]) <= 29:
                    average255[int(line[0]) - 1] += float(line[1])
          else:
                    pass


def normalize():
          for i in range(len(average0)):
          average0[i] *= (1 / 6)
          average100[i] *= (1 / 6)
          average150[i] *= (1 / 6)
          average200[i] *= (1 / 5)
          average255[i] *= (1 / 6)
          time.append(i * 4)


def graph():

          plot0 = mlines.Line2D(time, average0, color='blue', label='0 intensity', linewidth=4)
          plot100 = mlines.Line2D(time, average100, color='cyan', label='100 intensity',
linewidth=4)
          plot150 = mlines.Line2D(time, average150, color='green', label='150 intensity',
linewidth=4)
          plot200 = mlines.Line2D(time, average200, color='orange', label='200 intensity',
linewidth=4)
          plot255 = mlines.Line2D(time, average255, color='red', label='255 intensity',
linewidth=4)

          plt.legend(handles=[plot0, plot100, plot150, plot200, plot255], prop={'size': 15})

          plt.plot(time, average0, color='blue', linewidth=3)
          plt.plot(time, average100, color='cyan', linewidth=3)
          plt.plot(time, average150, color='green', linewidth=3)
          plt.plot(time, average200, color='orange', linewidth=3)
          plt.plot(time, average255, color='red', linewidth=3)
          title_font = {'fontname': 'Arial', 'size': '20', 'color': 'black', 'weight': 'normal',
                    'verticalalignment': 'bottom'}
```

```
        axis_font = {'fontname': 'Arial', 'size': '19'}
        plt.title('Distance between the worm and the motor in function of time', **title_font)
        plt.xlabel('Distance', **axis_font)
        plt.ylabel('Time in seconds', **axis_font)

        plt.show()


# calls
results0()
results100()
results150()
results200()
results255()
normalize()
graph()
```

We also did the data analysis for the electronic sensor. We plotted the intensity of the motor according to the value measured by the piezo at 25 cm, and then at 40 cm.
Below, the python code we created in order to gather the data, named **electronic_analysis.py**[12].

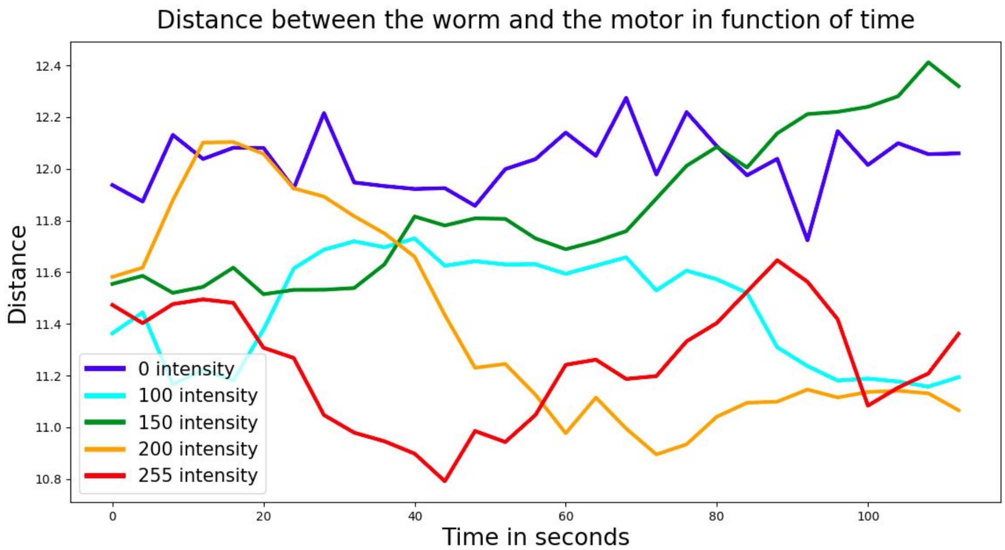
```
# coding = UTF-8
# written by Samuel Churlaud
# see github.com/learningthruresearch/Biosensors2017/tree/master/poworm_rangers
# version 2.1

# modules
import matplotlib.pyplot as plt

# variables
cleanData25 = {0: [], 100: [], 150: [], 200: [], 255: []}
cleanData40 = {0: [], 100: [], 150: [], 200: [], 255: []}
listVibrations = [0, 100, 150, 200, 255]
numberData = 3


# functions
def getData():

        for i in listVibrations:
        for j in range(numberData):
        with open('data/25_' + str(i) + '_' + str(j + 1), 'r') as rawData:
                for line in rawData:
```

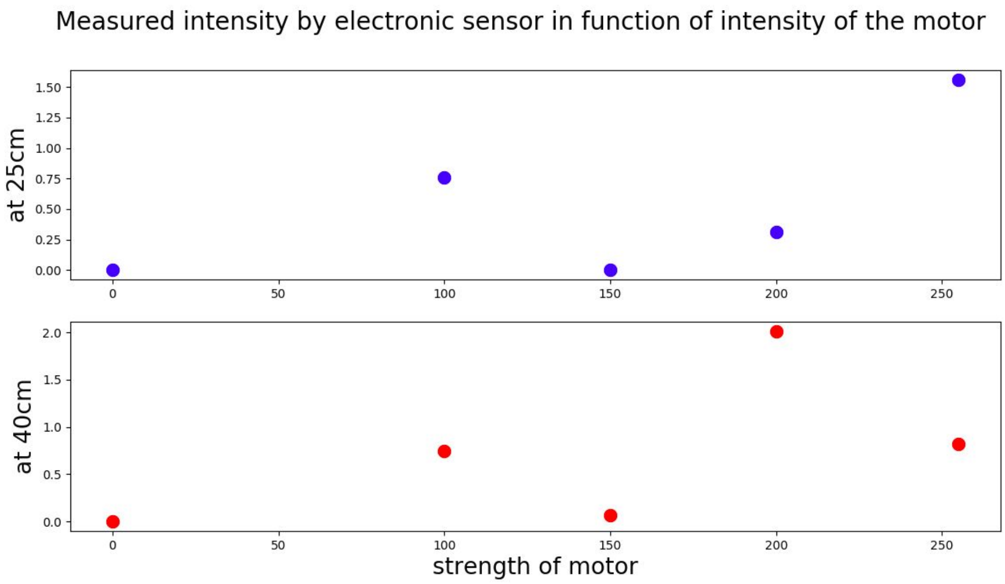
12 https://github.com/learningthruresearch/Biosensors2017/blob/master/poworm_rangers/code/electronic_analysis.py

```
                    key = i
                    cleanData25[key].append(line.strip('\n'))

          for i in listVibrations:
          for j in range(numberData):
          with open('data/40_' + str(i) + '_' + str(j + 1), 'r') as rawData:
                    for line in rawData:
                    key = i
                    cleanData40[key].append(line.strip('\n'))


def firstGraph():

          fig = plt.figure()
          fig.suptitle("electronic sensor")

          averageOne = []
          for i in cleanData25:
          values = 0
          for j in range(len(cleanData25[int(i)])):
          values += int(cleanData25[int(i)][int(j)])
          averageOne.append(values / (len(cleanData25[int(i)])))

          plt.subplot(211)
          # plt.title('average at 25cm')
          plt.ylabel('observed vibrations at 25cm')
          plt.plot(listVibrations, averageOne, marker='o', linestyle='none',
markeredgewidth=5, color='blue')

          averageTwo = []
          for i in cleanData40:
          values = 0
          for j in range(len(cleanData40[int(i)])):
          values += int(cleanData40[int(i)][int(j)])
          averageTwo.append(values / (len(cleanData40[int(i)])))

          plt.subplot(212)
          # plt.title('average at 40cm')
          plt.xlabel('strength of motor')
          plt.ylabel('observed vibrations at 40cm')
          plt.plot(listVibrations, averageTwo, marker='o', linestyle='none',
markeredgewidth=5, color='red')

          plt.show()


# calls
getData()
firstGraph()
```

Here the two graphs we obtained for electronic data and for biological data

The first graph shows the results obtained with biological sensor





Conclusions

If you want to discover more about our project:

- Consult our github, and gather all the material and documentation you need in order to reproduce our project
- Read the blogspot of the project
- See the presentation here
- Discover our twitter account and our storify
- Contact us