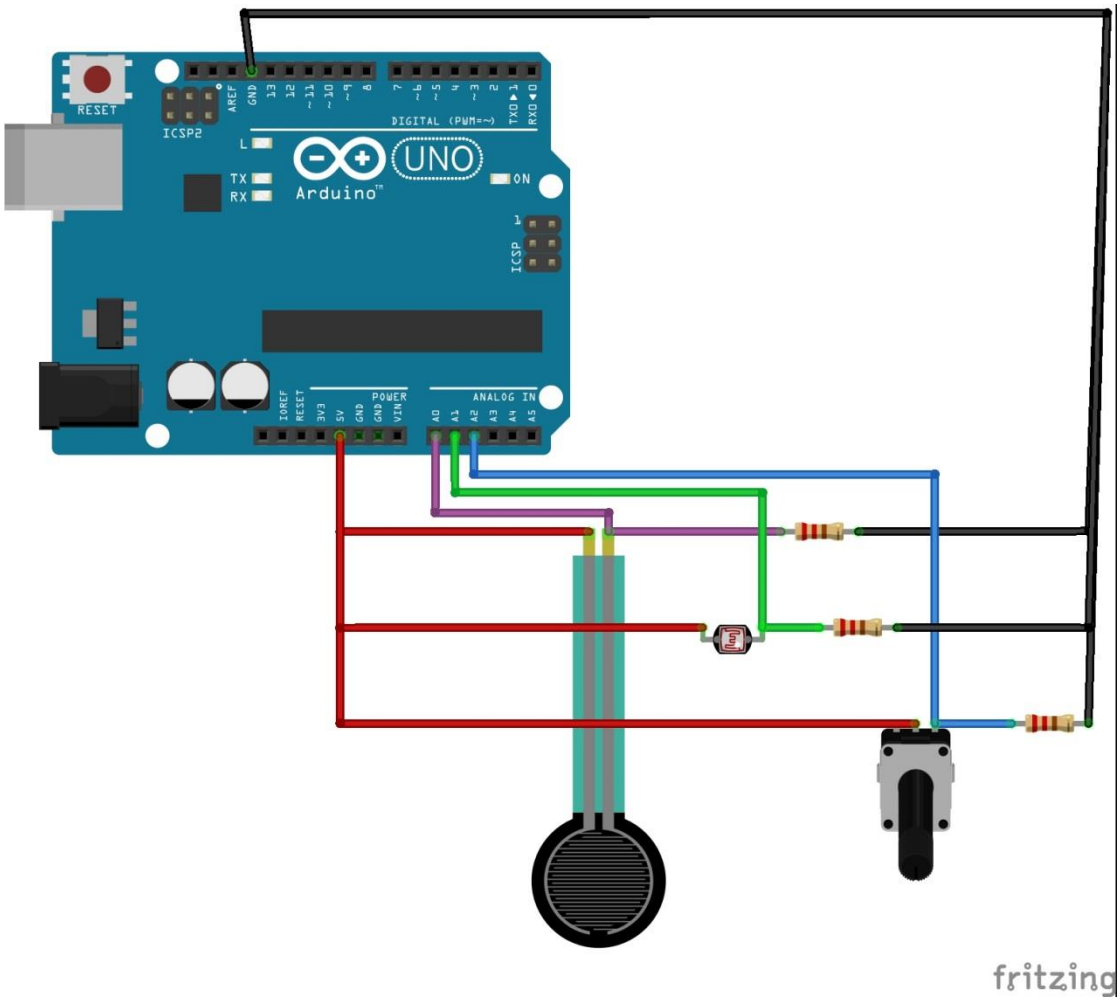
Acquisition de données analogiques à partir d'Arduino
Interface Python 2.7.XX

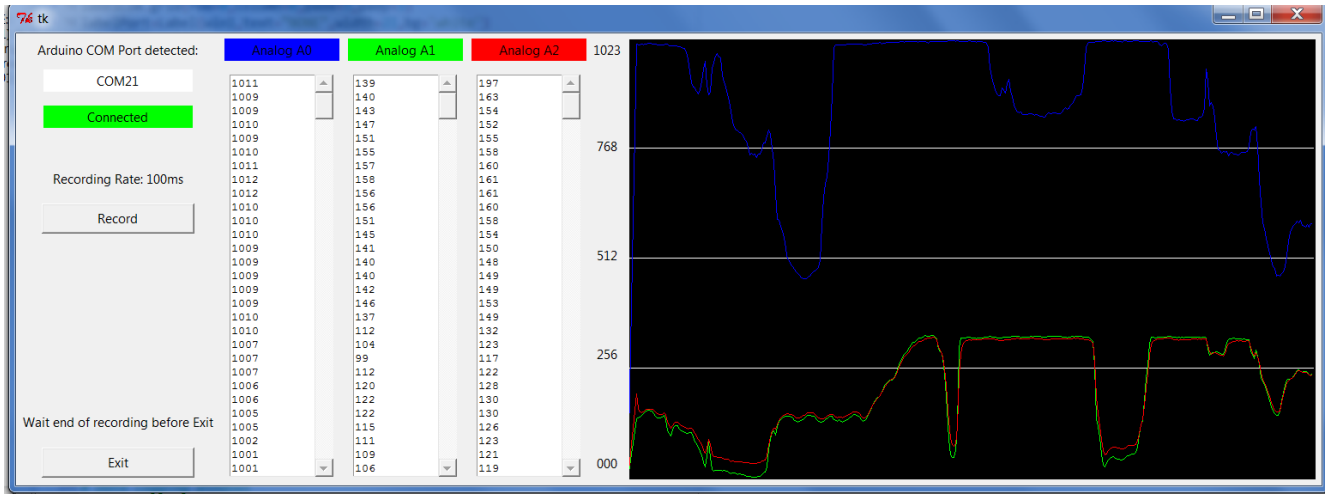
1. Câblage :

Montage basique avec par exemple ici un capteur de pression sur Analog A0, une photorésistance sur Analog A1 et un potentiomètre sur Analog A2. Les résistances fixes ajoutées en série aux capteurs permettent d'affiner la sensibilité des mesures en fonction de la plage de valeurs visée, pas de règle absolue, il faut essayer...



2. Interface Python :

L'interface utilisateur a été codée sous Python 2.7.6.1, elle détecte automatiquement le port de connexion d'Arduino et permet de voir les valeurs numériques des 3 entrées analogiques (qui peuvent être copiées ensuite), et d'avoir une visualisation graphique temps réel.



Lorsque l'Arduino a été détecté et connecté automatiquement, il suffit de cliquer sur « Record » et attendre la fin de l'enregistrement avant de quitter l'application.
Attention, pour relancer une nouvelle acquisition, il faut quitter puis redémarrer (je devrais corriger ça dans une prochaine version…).

Code Python

```
#--------------------------------------------------------------------------
-----
# Name:        Arduino 3 analog Channels data acquistion via serial COM Port
# Author:      Nicolas Blanchard
# Created:      06/04/2017
# Copyright:    (CC BY-NC-SA 4.0)
#--------------------------------------------------------------------------
-----

from Tkinter import *
from serial import *
from ScrolledText import *
import serial.tools.list_ports

# Global variables initialization
comPort='COM1'
a0Value,a1Value,a2Value=0,0,0
xstep=2
a0prValue,a1prValue,a2prValue=0,0,0
xst0,xfi0=0,10
timeDelay="0.1"

# Getting the COM Port list
ports = list(serial.tools.list_ports.comports())
for p in ports:
    print p

# Scaling the data for plotting
def scaleData(data):
    if data%2==0:
        return data/2
    else:
        return (data-1)/2

# Getting COM PORT for connection
def connectPort(event):
```

```
    global comPort
    comPort=entryCOM.get()
    print comPort
    labelConnected.configure(text="Connected",bg='Green')

# Plotting the lines with the recorded values
def recordData():
    global
xstep,a0Value,a1Value,a2Value,a0prValue,a1prValue,a2prValue,comPort,xst0,xf
i0,timeDelay

    with Serial(port=comPort, baudrate=9600, timeout=1, writeTimeout=1) as
port_serie:
        if port_serie.isOpen():
            while xfi0<800:
                ligne=str(port_serie.readline())
                if ligne!="":
                    if ligne[0]=="<":
                        a0Value=int(ligne[1:5])
                        textAnalogA0.insert(END,(str(a0Value)+'\n'))
                        textAnalogA0.update_idletasks()
                        lineA0= graphicPlot.create_line(xst0,512-
scaleData(a0prValue),xfi0,512-scaleData(a0Value),fill='blue')
                        a0prValue=a0Value
                        a1Value=int(ligne[6:9])
                        textAnalogA1.insert(END,(str(a1Value)+'\n'))
                        lineA1= graphicPlot.create_line(xst0,512-
scaleData(a1prValue),xfi0,512-scaleData(a1Value),fill='green')
                        a1prValue=a1Value
                        a2Value=int(ligne[10:13])
                        textAnalogA2.insert(END,(str(a2Value)+'\n'))
                        lineA2= graphicPlot.create_line(xst0,512-
scaleData(a2prValue),xfi0,512-scaleData(a2Value),fill='red')
                        a2prValue=a2Value
                        xst0=xfi0
                        xfi0=xfi0+xstep
                    else:
                        xfi0=xfi0

                print "AO:",2*int(a0Value),"-A1:",2*int(a1Value),"-
A2:",2*int(a2Value)

# Main program
win1=Tk()

# Com Port Connection
labelCom=Label(win1,text="Arduino COM Port detected:")
labelCom.grid(row=0,column=0,padx=5,pady=3)
labelPort=Label(win1,text="NONE",width=21,bg='white')
labelPort.grid(row=1,column=0,padx=5,pady=1)
labelConnected=Label(win1,text="Disconnected",bg='Red',width=21)
labelConnected.grid(row=3,column=0,padx=5,pady=1)

ports = list(serial.tools.list_ports.comports())
for p in ports:
    if p[1].find('Arduino')!=-1:
        comPort=p[0]
        print comPort
        labelPort.configure(text=comPort)
        labelConnected.configure(text="Connected",bg='Green')
```

```
##entryCOM=Entry(win1,width=21)
##entryCOM.grid(row=1,column=0,padx=5,pady=1)
##entryCOM.bind("<Return>",connectPort)


# Recording Rate
labelRate=Label(win1,text="Recording Rate: 100ms",width=21)
labelRate.grid(row=5,column=0,padx=5,pady=5)

# Record launching
buttonRecord=Button(win1,text="Record",width=21,command=recordData)
buttonRecord.grid(row=6,column=0,padx=5,pady=5)

# Data logging windows
labelAnalogA0=Label(win1,text="Analog A0",width=16,bg='blue')
labelAnalogA0.grid(row=0,column=1,padx=5,pady=3)
labelAnalogA1=Label(win1,text="Analog A1",width=16,bg='green')
labelAnalogA1.grid(row=0,column=2,padx=5,pady=3)
labelAnalogA2=Label(win1,text="Analog A2",width=16,bg='red')
labelAnalogA2.grid(row=0,column=3,padx=5,pady=3)
textAnalogA0=ScrolledText(win1,width=12,height=29,bg='white')
textAnalogA0.grid(row=1,column=1,rowspan=29)
textAnalogA1=ScrolledText(win1,width=12,height=29,bg='white')
textAnalogA1.grid(row=1,column=2,rowspan=29)
textAnalogA2=ScrolledText(win1,width=12,height=29,bg='white')
textAnalogA2.grid(row=1,column=3,rowspan=29)

# Graphic window design and scales
graphicPlot=Canvas(win1,width=800,height=512,bg='black')
graphicPlot.grid(row=0,column=6,rowspan=29,padx=5,pady=5)
line1= graphicPlot.create_line(0,128,800,128,fill='white')
line2= graphicPlot.create_line(0,256,800,256,fill='white')
line3= graphicPlot.create_line(0,384,800,384,fill='white')

labelScale1=Label(win1,text='1023')
labelScale1.grid(row=0,column=5)
labelScale2=Label(win1,text='768')
labelScale2.grid(row=4,column=5)
labelScale3=Label(win1,text='512')
labelScale3.grid(row=7,column=5)
labelScale4=Label(win1,text='256')
labelScale4.grid(row=20,column=5)
labelScale5=Label(win1,text='000')
labelScale5.grid(row=28,column=5)

# Exit Button
labelExit=Label(win1,text='Wait end of recording before Exit')
labelExit.grid(row=27,column=0,padx=5,pady=5)
buttonExit=Button(win1,text="Exit",width=21,command=win1.destroy)
buttonExit.grid(row=28,column=0,padx=5,pady=5)

win1.mainloop()
```

3. Code Arduino :

Rien de bien révolutionnaire, hormis que les valeurs des entrées analogiques sont écrites dans un seul « mot » entre balises qui ressemble à <100100650189> qui veut dire valeur de A0 : 1001, valeur de A1 : 0065, valeur de A2 : 0189.

Code Arduino

```
/*****************************************************************
ANALOG DATA ACQUISITION VIA SERIAL FOR 3 CHANNELS - PYTHON
*****************************************************************
Written by Nicolas Blanchard - (CC BY-NC-SA 4.0)
*****************************************************************/

// When DEGUG is TRUE send an newline to the serial monitor
const boolean DEBUG = true;

const byte A0Pin = A0;
const byte A1Pin = A1;
const byte A2Pin = A2;

int RecordingRate = 100;

unsigned int newA0Val = 0;
unsigned int newA1Val = 0;
unsigned int newA2Val = 0;

// used to hold an ascii representation of a number
char numberString[10];

void setup()
{
  // open serial communication
  Serial.begin(9600);
  Serial.println("Adruino is ready");
  Serial.println(" ");
}

void loop()
{
  // get the recording rate value
  if (Serial.available() > 0) {
    RecordingRate = Serial.parseInt();
    Serial.println(RecordingRate);
  }
  // read the pins
  newA0Val = analogRead(A0Pin);
  newA1Val = analogRead(A1Pin);
  newA2Val = analogRead(A2Pin);
  formatNumber( newA0Val, 4);
  Serial.print("<");
  Serial.print(numberString);
  formatNumber( newA1Val, 4);
  Serial.print(numberString);
  formatNumber( newA2Val, 4);
  Serial.print(numberString);
  Serial.print(">");
  if (DEBUG) {
    Serial.println("");
  }
  delay (RecordingRate);
}

void formatNumber( unsigned int number, byte digits)
{
  // formats a number in to a string and copies it to the global char array
numberString
```

```
  // pads the start of the string with '0' characters
  //
  // number = the integer to convert to a string
  // digits = the number of digits to use.

  char tempString[10] = "\0";
  strcpy(numberString, tempString);

  // convert an integer into a acsii string
  itoa (number, tempString, 10);

  // create a string of '0' characters to pad the number
  byte numZeros = digits - strlen(tempString) ;
  if (numZeros > 0)
  {
    for (int i = 1; i <= numZeros; i++)    {
      strcat(numberString, "0");
    }
  }
  strcat(numberString, tempString);
}
```