Ignacio Sánchez Aparicio y Alejandro Mayor Escalada

PRIMERA VERSIÓN OPERATIVA

Código Arduino:

#define PINCANAL\_A 0

void setup()

{

Serial.begin(9600);

}

void loop()

{

int value =analogRead(PINCANAL\_A);

value = (value » 8) & 0xff; //solo cogeremos el valor de los últimos 8 bits, el resto los desechamos

Serial.write(value & 0xff);

}

Código Processing: (no terminado)

import processing.serial.\*; //importamos la librería serial

Serial myPort; //nombre del puerto serie

int value; //desde el puerto serie

int[] valores;

float zoom;

void setup() //Función de arranque

{

size(1600, 720); //Tamaño de la ventana en el que se dispondrá la gráfica

port = new Serial(this, “COM 3”, 9600); //Se abre el puerto de serie COM3

valores = new int(width);

int valorbyte() { //Esta función devuelve un número entre 0 y 255 que se refiere al próximo byte esperando en el buffer

valor = -1;

while (myPort.available() > 0) { //En primer lugar comprobamos si hay espacio libre

if(myPort.read() == 0xff){

valor= myPort.read(<<8) | (myPort.read()); //Al igual que en el código de arduino cogemos ignoramos los bytes que no sean los últimos ocho

}

return valor;

}

int coordenadaY(int value) { //Transformamos value que nos llega desde el puerto para representarlo en el eje y

return (int)(height - value / 1023.0f \* (height - 1));

}

void drawLines() { //Representamos el gráfico con esta función

stroke(200); //Color de los bordes

int k = values.length - width;

int x0 = 0;

int y0 = coordenadaY(valores[k]);

for (int i=1; i<width; i++) {

k++;

int x1 = (int) (i \* (width-1) / (width-1));

int y1 = getY(values[k]);

line(x0, y0, x1, y1); //Se dibujará una recta entre los puntos X0 e Y0

x0 = x1;

y0 = y1;

}

}