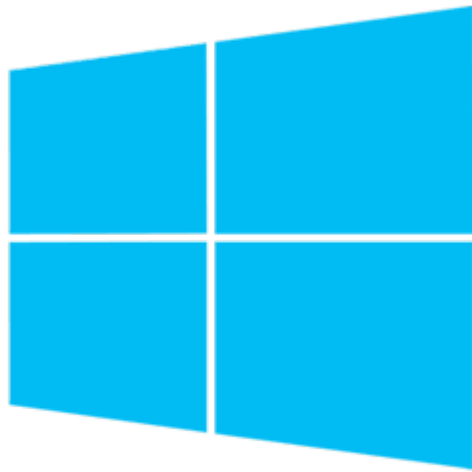


# Serial Sensor Plot

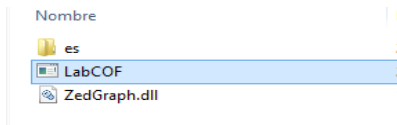
## Version 1.0



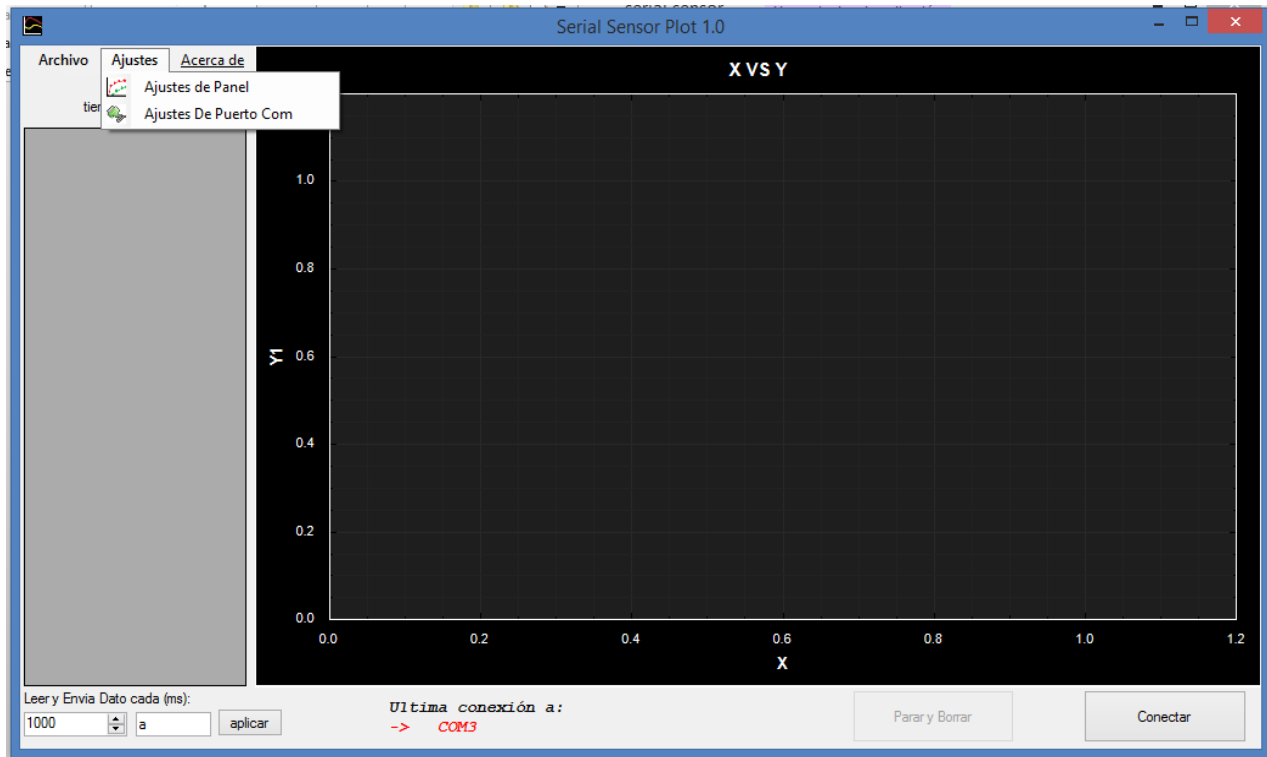
Windows®

# Conexión rápida cuando solo se quiere graficar un solo dato

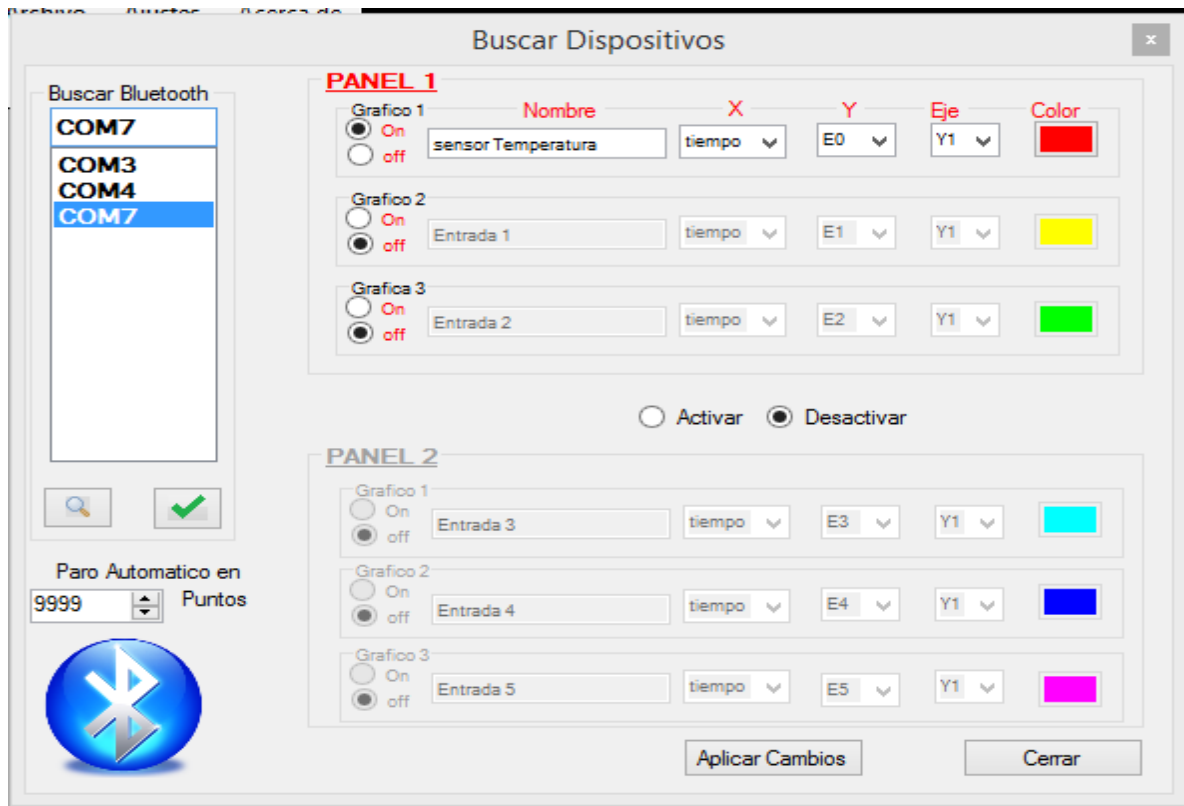
Una vez descargada la carpeta “serial sensor” OJO “son necesarios tener los tres archivos en la misma carpeta”



Abrimos el archivo “LabCOF” lo siguiente es conectarse al puerto com donde esta nuestro arduino vamos a ajustes-Ajustes De Puerto Com.



se nos abrirá la siguiente ventana



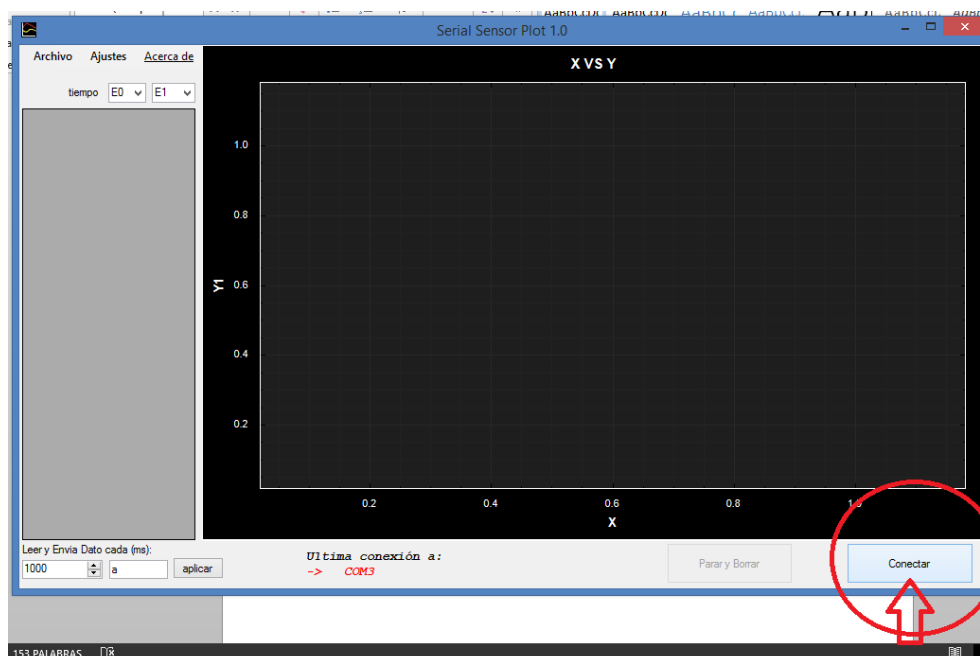
Damos click en la Lupa para buscar los puertos com existentes. Elegimos nuestro puerto com dando clic en el botón . Por default el software trabaja con un solo dato que debe ser enviado con espacio al final así: **"23.45 "** y este se graficara en el panel 1.

En grafico 1 elegimos un nombre para nuestra serie de puntos que después podremos visualizar en el gráfico. He colocado el nombre "sensor Temperatura".

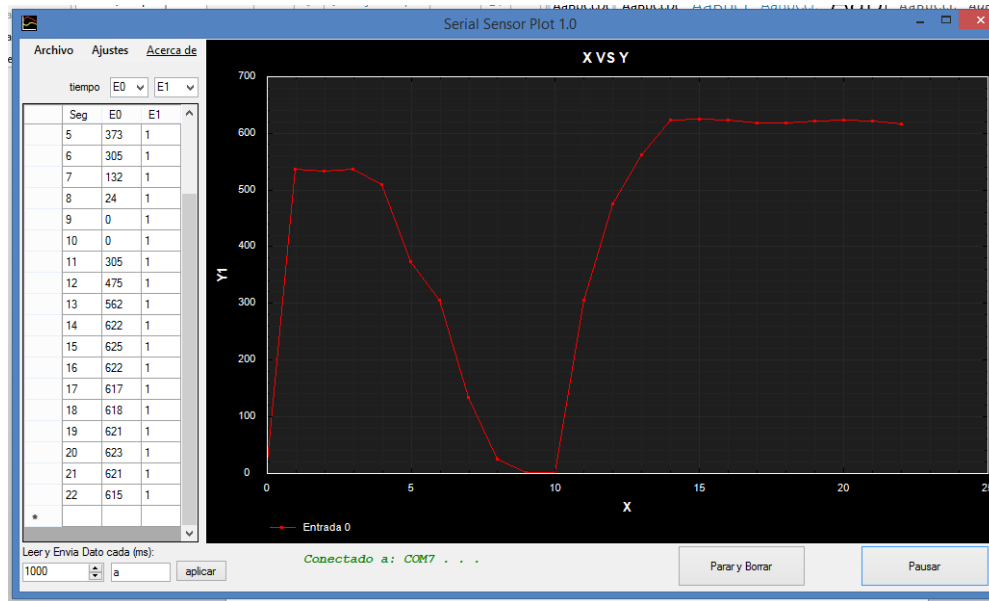
Por default se grafica el tiempo en el eje X y nuestro dato en Y que llamamos E0 y será de color Rojo. El color así como el eje donde queremos graficar cada dato son intercambiables.

Por ultimo damos "Aplicar cambios."

Ahora en nuestro panel de graficado damos clic en conectar.



Si nuestro arduino ya está enviando datos debemos de comenzar a ver cómo se van pintando estos.



Por default las lecturas se hacen cada 1seg=1000ms, y se envía el carácter “a” estos son modificables quizá menos de 100ms pueda provocar error sobre todo si se tiene más de 1 dato graficando.

En el panel izquierdo podemos ver nuestro tiempo cada vez que se hace una lectura al puerto serie y el respectivo dato que este nos entrega. Existe una tercer columna que sirve para visualizar otro dato de los 9 posibles que podemos recibir.

## Personalización del panel de graficado.

Click en ajustes-Ajustes de Panel y se nos abrirá la siguiente ventana.

**-Panel1:** En títulos podemos elegir el “título del grafico” el “nombre de las variables” o datos en los eje X,Y1,Y2. (ojo el eje Y2 se pinta en el lado derecho útil cuando tenemos otra escala u otra unidad de medida)

- Escalado:** al ir llegando más puntos podemos ir auto-escalando el panel o simplemente fijar los límites del panel.
- **Panel2:** es para personalizar el panel 2 cuando estamos usando dos paneles (el panel 2 se activa en ajustes de puesto com)
- **Colores:** podemos cambiar los colores del fondo, de la cuadrícula, de los títulos de los ejes.
- **puntos:** podemos elegir el tamaño del punto el tipo de punto y si estos se unen con una línea o si solo se muestran puntos sin unión.

Por ultimo Puedes guardar los datos que estén graficados en pantalla en formato excel dando clic en Archivo guardar(ojo es necesario tener instalado al menos office 2003 o superior).

## Graficas múltiples en uno y dos paneles

Primeramente es necesario que nuestro arduino o cualquier microcontrolador este enviando los datos de la siguiente manera;

Dejando 1 espacio al final de cada dato, no olvidar el espacio final. Así:

**“3.45 56.78 2.0 45.67 78.90 ” aquí hay 5 datos como máximo pueden ser nueve (la app para android tomara como maximo 5)**

**“3.45 56.78 2.0 45.67 78.90 3.4 5.78 55.33 8.9 ” aquí hay nueve datos (ojo espacio al final)**

El software parte la cadena y extrae cada valor (no debe haber texto solo números o marcara error) los datos son enumerados como E0,E1E2,E3....E8 que son como llegan en la cadena. Cada uno de estos datos pueden provenir de un sensor como el de temperatura humedad presión punta hall etc..

A continuación se da como ejemplo código en arduino que puede ser usado. El siguiente código solo envía dos datos del pulso cardiaco en el pin8 y un potenciómetro que tiene colocado en el puerto A0 que es analógica.

PirPulso \$

```
byte  sensorCardiaco = 8;//entrada de arduino donde esta el sensor cardiaco
String envia="";//string (vacio al inicio) que tendra toda la cadena a enviar con cada dato incrustado
      //y separado por un espacio entre cada dato y al final

void setup()
{
  //definimos como entrada, configuramos el puerto serie a 9600 que es el que usa "Serial Sensor Plot 1.0"
  pinMode(sensorCardiaco,INPUT);
  pinMode(A0,INPUT);
  Serial.begin(9600);
}
void loop()
{
  // se carga el valor analogico leido en A0 del arduino al strin "envia" sin dejar espacio al inicio
  envia.concat(analogRead(A0));
  //agregamos un espacio a la cadena anterior
  envia.concat(" ");
  // leemos el pin 8 para ver si hay pulso si lo hay pegamos al strin "envia" un 1 de lo contrario le pegamos un 0
  if(digitalRead(sensorCardiaco)==HIGH){envia.concat(1);} else {envia.concat(0);}
  //agregamos nuevamente un espacio
  envia.concat(" ");
  //si quisieramos agregar otro dato lo hacemos usando "envia.concat(DATO)" y no olvidar nuevamente el
  //espacio al final envia.concat(" ");

  //enviamos
  Serial.println(envia);
  //limpiamos el string "envia" para iniciar nuevamente el loop
  envia="";
  //un retardo aveces si no lo pongo me causa problemas
  delay(1);
}
```

Subido

Si nuestro arduino ya está enviando más de 1 datos ahora vamos a ajustes-Ajustes De Puerto Com.

En **panel 1** activamos grafico 2 colocamos un nombre yo le he llamado “pulso cardiaco” por default en x nuevamente está el tiempo de lectura, E1 automáticamente es tomado del segundo dato de la cadena que envía arduino y es mostrado en el eje izquierdo Y1 puesto que nuestro potenciómetro dará valores entre 0 y 1024 y nuestro pulso cardiaco solo 1 o 0. Podemos elegir visualizar este dato en el eje Y2 o sea al lado derecho del panel 1. Podemos también elegir el color del gráfico. O este es amarillo y el anterior dato rojo.

Nota: no es necesario enviar los 9 datos puedes enviar 1 o 7 o 4, es importante que el delay() de arduino coincida con el del software algual que el de la app si es que usas android. El codigo de arduino es totalmente compatible con el soft para windows como para android.

### Buscar Dispositivos

#### Buscar Bluetooth

Paro Automatico en

9999

Puntos

#### PANEL 1

Grafico 1
Nombre
X
Y
Eje
Color

☒ On  
☐ off

tiempo

▼

E0

▼

Y1

▼

☒ On  
☐ off

tiempo

▼

E1

▼

Y2

▼

☐ On  
☒ off

tiempo

▼

E2

▼

Y1

▼

☐ Activar    ☒ Desactivar

#### PANEL 2

☐ On  
☒ off

tiempo

▼

E1

▼

Y1

▼

☐ On  
☒ off

tiempo

▼

E4

▼

Y1

▼

☐ On  
☒ off

tiempo

▼

E5

▼

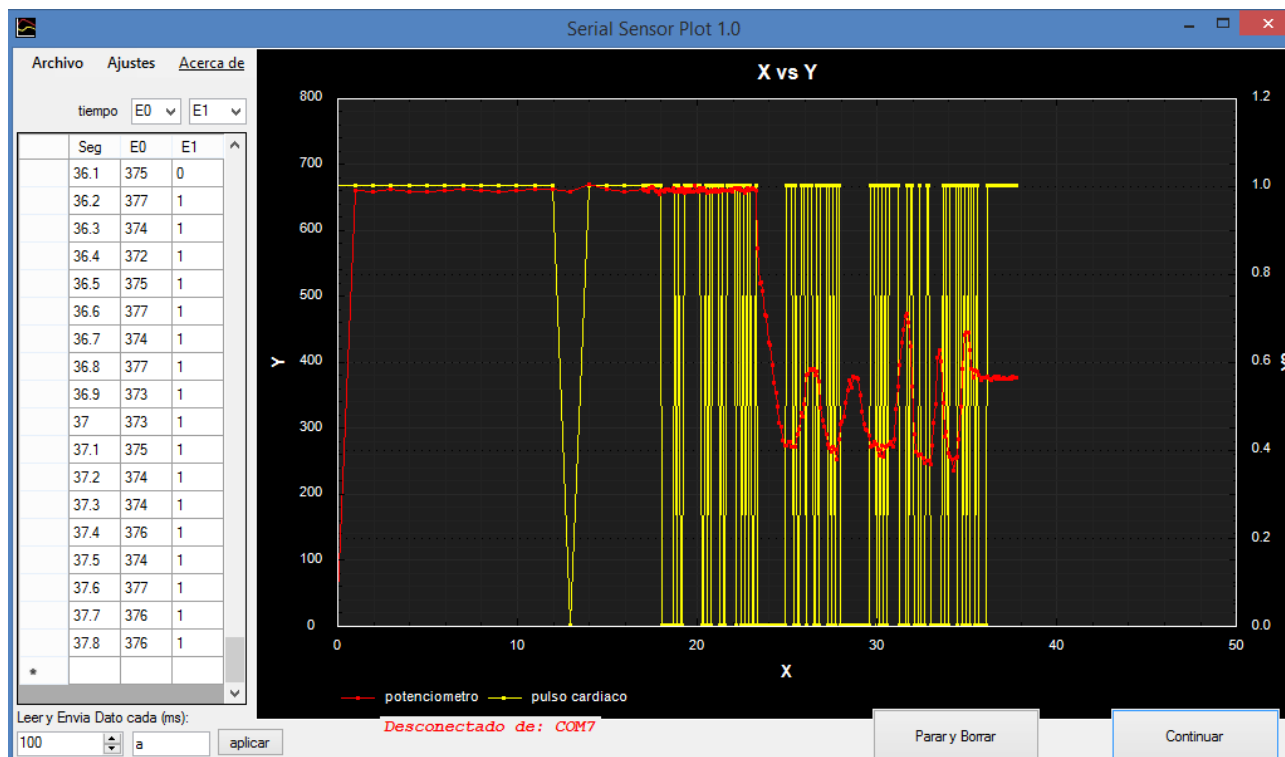
Y1

▼

Aplicar Cambios

Cerrar

Damos aplicar cambios y deberíamos observar algo así;



Por ultimo podemos cambiar el tirulo de los ejes en Ajustes-Ajustes de panel.

### Ajustes De Grafico

**PANEL 1**  
 Titulos  
 Principal:   
 Eje X:   
 Eje Y1:   
 Eje Y2:

Escalado  
☒ Automatico ☐ Definir  
 Eje X Eje Y  
 Min.    
 Max.

**PANEL 2**  
 Titulos  
 Principal:   
 Eje X:   
 Eje Y1:   
 Eje Y2:

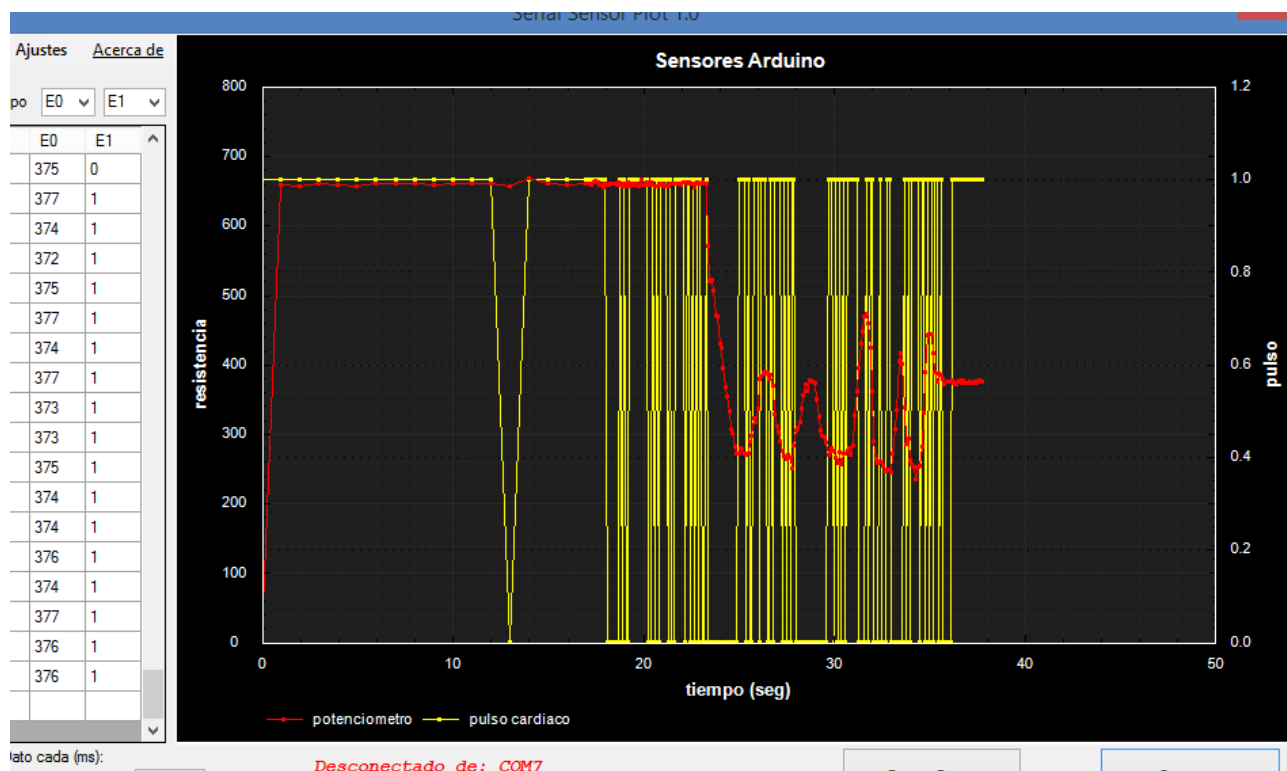
Escalado  
☒ Automatico ☐ Definir  
 Eje X Eje Y  
 Min.    
 Max.

Colores  
 Fondo1 Fondo2 Cuadrícula Titulo Ejes  
    X Y2  
   Y1

Puntos  
 Tamaño Tipo de Punto Unir con Línea  
 2.0 Circle Si

☐ Reset Ajustes

Damos aplicar cambios y obtenemos lo siguiente, los títulos han cambiado.

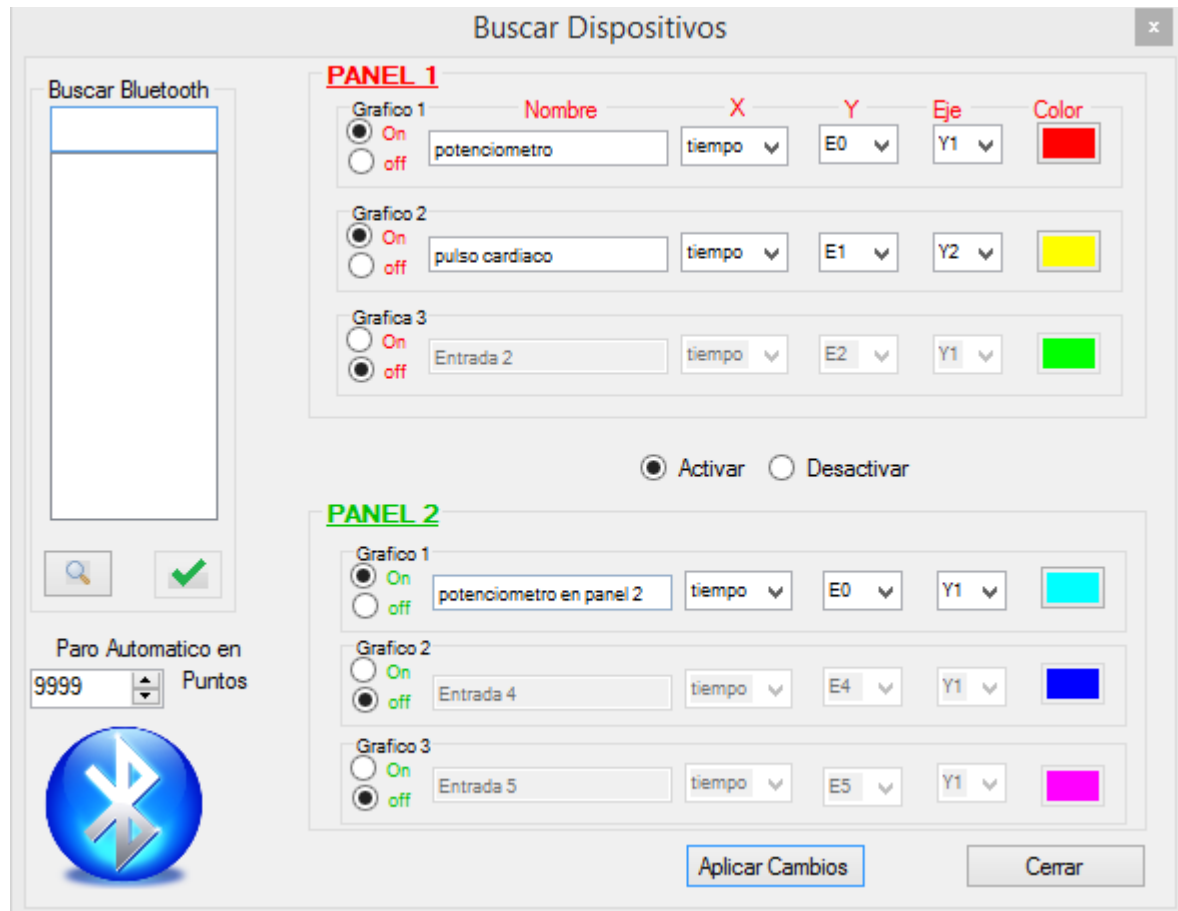




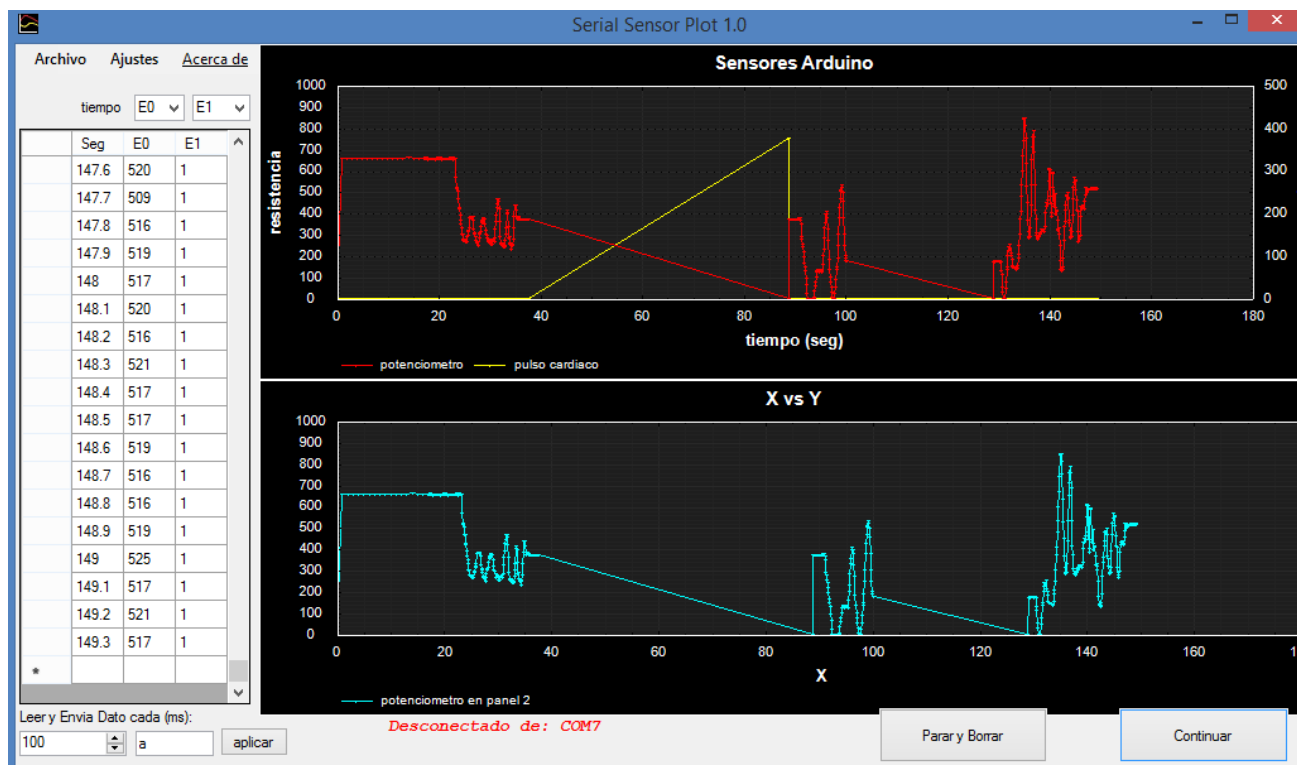
## Como usar el segundo panel?

Vamos a Ajustes-Ajustes De Puerto Com

Y activamos el panel 2 y podemos nuevamente elegir otros 3 graficos posibles elegimos grafico 1 en panel 2 y graficamos nuestros datos del potenciómetro así (no olvidar aplicar cambios);



Y listo tenemos un segundo panel que si le damos conectar tenemos lo siguiente;



Como podemos observar el grafico rojo se repite en el panel 2.

Si bien solo podemos tener 6 graficas en pantalla ya sea todas en el panel1 o compartidas con el panel 2 cabe señalar que podemos enviar hasta 9 datos podríamos cambiar entre un sensor y otro según se desee.

Incluso podríamos graficar un sensor VS otro es muy útil si deseamos ver la dependencia de 2 variables como por ejemplo dilatación según la temperatura de hecho originalmente fue creado este soft para ese experimento.

Cabe señalar que el programa tiene como máximo 9999 puntos y posee un paro automático en la cantidad de puntos señalada en Ajustes-Ajustes de puerto Com. Esto es útil si se hace un experimento que tome datos por horas está pensado para tomar datos cada 5 segundos por 12 horas alcanzaría cerca de los 9999 puntos totales. Esto es modificable desde el código fuente. Si no se dese paro automático lo más recomendado es dejar en 9999.

Espero a alguien les sea útil este software es una primer versión es obvio que faltan mas cosas por agregar pero aun así resulta bastante útil para graficar varios datos al mismo tiempo.

---

Todo el código esta en github. <https://github.com/johnspice/Serial-Graph-Sensor>

Solo se requiere de visual estudio la última versión y eso es todo puedes modificarlo.

Si te sirvió este software y deseas contribuir a mi trabajo baja la aplicación para Android que tiene la misma función. <https://play.google.com/store/apps/details?id=com.troyasoft.pro.serialplot>

Este proyecto a dado origen a la creacion de la libreria "Jplot" para android de libre uso ya se trabaja en una version de la misma para IOS con swift. <https://github.com/johnspice/jplot-android>

# Serial Sensor Plot Version 2.0



ANDROID

1.- Antes de abrir la aplicación debe enlazarse el modulo bluetooth del arduino con el celular o Tablet

La aplicación se probó con los módulos hc-05 y hc-06.

En Ajustes-bluetooth. (una forma de darse cuenta que se ha enlazado correctamente es que el led de modulo hc-05 o hc-06 deja de parpadear rápido.)

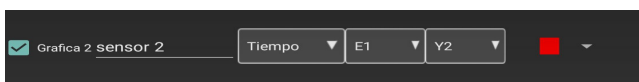
2.- abrir la aplicación dar click en el botón Buscar, si se enlazo correctamente en el paso 1 debera aparecer el bluetooth del cual recibiremos datos.

3.- click en Ajustes



**Los primeros 4 campos son los títulos que se mostraran en el gráfico elegirlos al gusto.**

Las siguientes 5 secciones representan la configuración de las 5 posibles gráficas que se pueden mostrar en tiempo real.



\*al seleccionar el checkbox se activa la grafica para poderse configurar.

\*el primer campo debe darse tener un nombre a la grafica.

\*el siguiente campo es un menu desplegable donde elegiremos que “dato X” por default esta configurado el tiempo en seg, pero puede ser cualquiera de los 5 que se recibirán.

-nota: si se elige un dato que no esta llegando marcara error al graficar. Asegurase que el dato elegido se esta enviando desde el circuito.

Si se recibe “E0 E1 E2 ” y eliges E3 marcara error.

\*el siguiente menu desplegable elegiremos que “dato Y” graficaremos puede ser cualquiera de los que se envian. Para formar la pareja de puntos (DatoX, DatoY).

-nota: si se elige un dato que no esta llegando marcara error al graficar. Asegurase que el dato elegido se esta enviando desde el circuito.

\*el siguiente menu permite elegir que “eje y” se usara (Y1 izquierda, Y2 derecha).

\*el ultimo menu desplegable nos permite asignarle un color al gráfico.

Al final hay un campo que dice “paro automático” nos permite parar automáticamente la lectura después de n puntos. El máximo a recabar es de 10000 lecturas o puntos en la gráfica, el mínimo es 1. No debe sobrepasarse esa cantidad. Esta limitante es debido a problemas con el rendimiento en algunos dispositivos de gama baja. Si se van usar las 5 gráficas es recomendable no tomar mas de 2000 lecturas, mas de esto dependiendo del dispositivo podría provocar realentizaciones.

Por ultimo dar click en “guardar ajustes” o cancelar.

Si ya aparece el dispositivo al que se conectara se debe seleccionar y dar click en conectar, si todo salió bien debería montarse una pantalla como la siguiente.

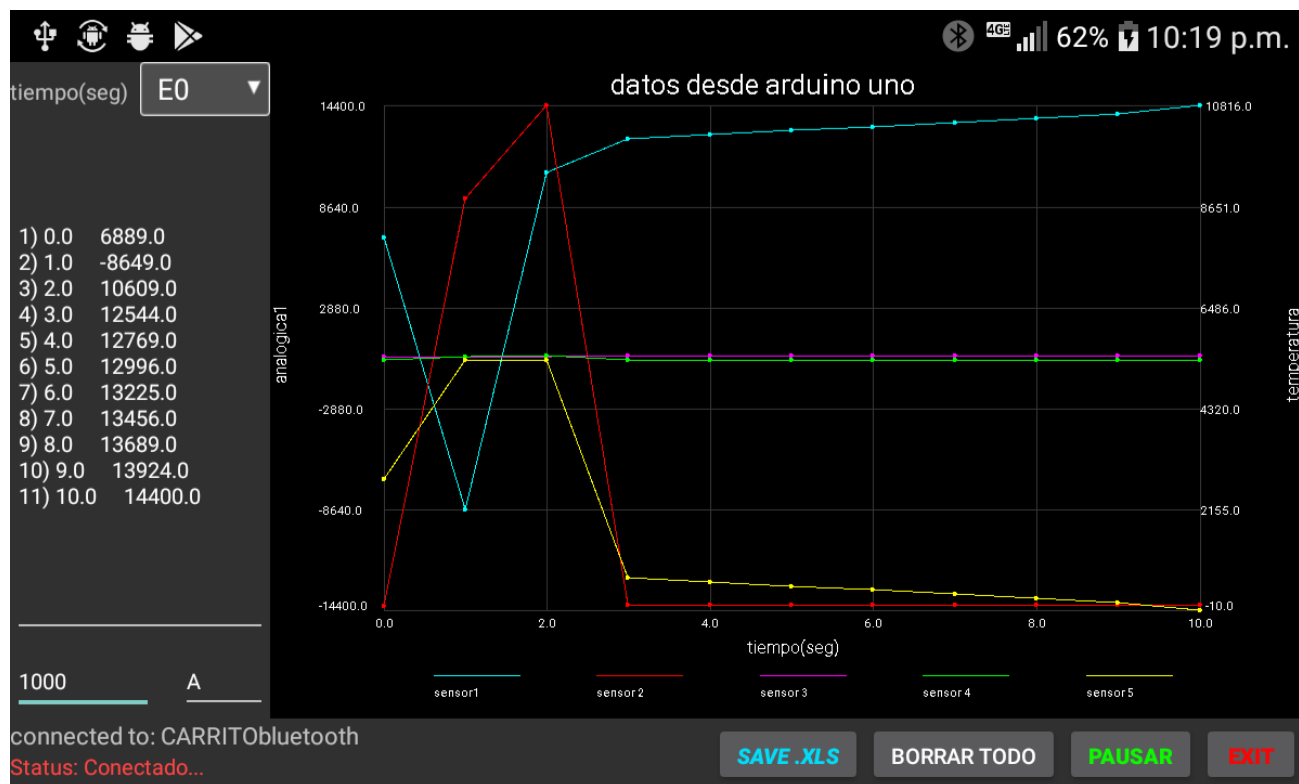
#### **4.- graficando los datos en tiempo real**

Al igual que en el software para windows aqui tenemos una interfaz de usuario mas o menos igual.

\*En la parte izquierda inferior hay 2 campos editables en los que podremos ajustar el tiempo de lectura y el dato que enviaremos en cada lectura.

**Nota: es importante que el tiempo(milisegundos) que elijas sea exactamente igual al delay() del Arduino(o del dispositivo serial que envíe los datos) en cada loop. De preferencia no deben elegirse tiempos menores a los 200ms, para evitar problemas de rendimiento.**

El dato que se envia por default siempre es “A” la intención de poder enviar datos es que se pueda tener algún control del tipo de envio del circuito.



\*en la parte superior izquierda hay un menu desplegable que puede mostrar 1 dato de entrada. Debe tenerse cuidado de no elegir un dato que no se recibe o marcara error.

\*se tiene el boton "Save.xls" al dar click pausara la lectura y guardara los datos que se esten graficando en ese momento en un archivo excel 1997-2003. en la carpeta "Serial Sensor" en la memoria interna del dispositivo.

- ! "Borrar Todo" permite limpiar los datos de entrada y el panel de graficado para volver a iniciar la lectura.
- ! "Pausar" pausa la lectura y tambien la reanuda.
- ! "Exit" detendra la lectura y volvera a la pantalla anterior.

## Código de prueba de Arduino

---

Este código sirve para windows y para Android

```
String envia = "";
int i = 0;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
}
void loop() {
    // put your main code here, to run repeatedly:
    envia.concat(i * i);
    envia.concat(" ");
    envia.concat(i);
    envia.concat(" ");
    envia.concat(i + 20);
    envia.concat(" ");
    envia.concat(-i);
    envia.concat(" ");
    envia.concat(-i * i);
    envia.concat(" ");
    Serial.println(envia);
    envia = "";
    i = i + 1;
    delay(1000); //debe ser el mismo que en la aplicacion ya sea android o windows
}
```

Nota: es muy importante usar `.concat`, nunca debes enviar texto, el espacio al final siempre debe ir aunq sea 1 solo dato. El espacio entre cada dato es importante. Recuerda que el tiempo debe ser el mismo que se debe ajustar en la app de android o del soft para windows.

## Código caso real de Arduino

### Este código sirve para windows y para Android

PirPulso \$

```
byte sensorCardiaco = 8;//entrada de arduino donde esta el sensor cardiaco
String envia="";//string (vacio al inicio) que tendra toda la cadena a enviar con cada dato incrustado
//y separado por un espacio entre cada dato y al final

void setup()
{
    //definimos como entrada, configuramos el puerto serie a 9600 que es el que usa "Serial Sensor Plot 1.0"
    pinMode(sensorCardiaco,INPUT);
    pinMode(A0,INPUT);
    Serial.begin(9600);
}

void loop()
{
    // se carga el valor analogico leído en A0 del arduino al strin "envia" sin dejar espacio al inicio
    envia.concat(analogRead(A0));
    //agregamos un espacio a la cadena anterior
    envia.concat(" ");
    // leemos el pin 8 para ver si hay pulso si lo hay pegamos al strin "envia" un 1 de lo contrario le pegamos un 0
    if(digitalRead(sensorCardiaco)==HIGH){envia.concat(1);} else {envia.concat(0);}
    //agregamos nuevamente un espacio
    envia.concat(" ");
    //si quisieramos agregar otro dato lo hacemos usando "envia.concat(DATO)" y no olvidar nuevamente el
    //espacio al final envia.concat(" ");

    //enviamos
    Serial.println(envia);
    //limpiamos el string "envia" para iniciar nuevamente el loop
    envia="";
    //un retardo aveces si no lo pongo me causa problemas
    delay(1);
}
```

Subido

**Nota: es muy importante usar .concat, nunca debes enviar texto, el espacio al final siempre debe ir aunq sea 1 solo dato. El espacio entre cada dato es importante. Recuerda que el tiempo debe ser el mismo que se debe ajustar en la app de android o del soft para windows.**