



# Índice

<b>Introducción.....</b>	<b>3</b>
<b>Descripción del proyecto .....</b>	<b>4</b>
<b>Realización del proyecto.....</b>	<b>5</b>
<b>Materiales.....</b>	<b>12</b>
<b>Resultados y valoración.....</b>	<b>12</b>
<b>Bibliografía y fuentes .....</b>	<b>12</b>

## Introducción

Este proyecto está enfocado a la recogida de datos, en este caso la temperatura, y su posterior tratado y visualización en pantalla y así poder hacer un seguimiento de la temperatura obtenida en el entorno.

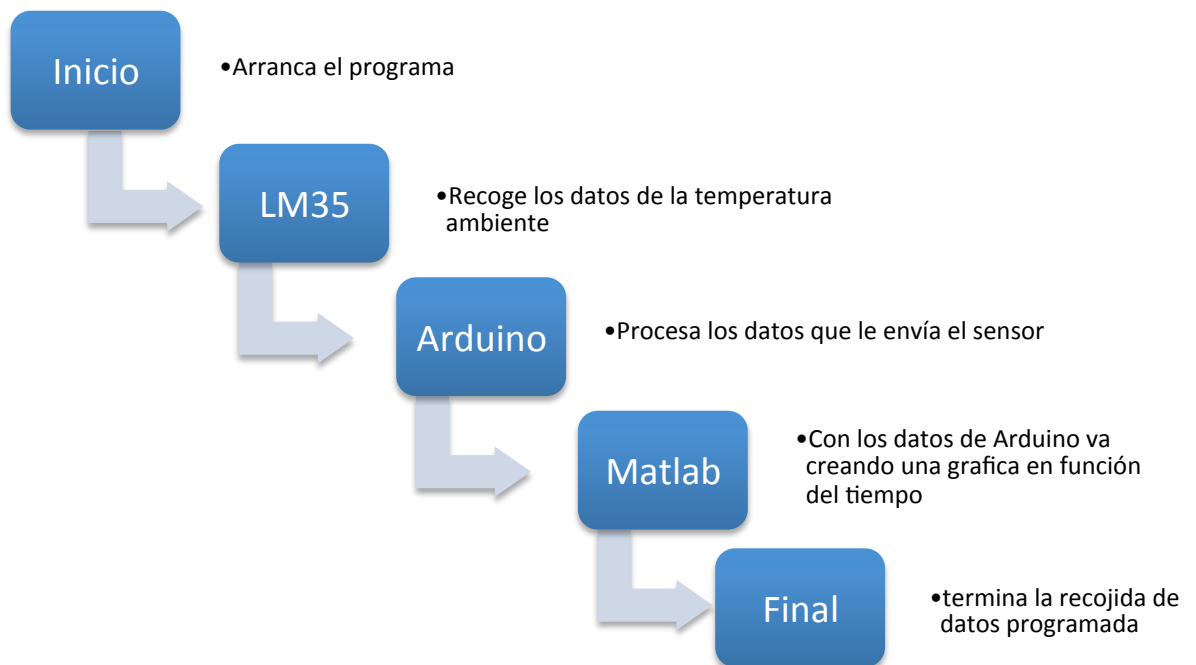
Para realizarlo hemos dispuesto de un sensor de temperatura **LM35** que toma las medidas del entorno.

Este está conectado a una placa de **Arduino** que cuenta con un programa por el cual recoge los datos del sensor de temperatura y los transforma en grados centígrados.

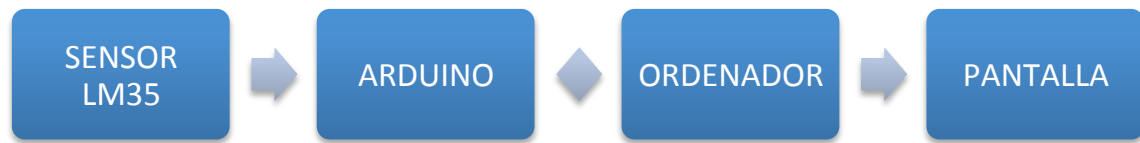
Este **Arduino** está conectado al ordenador y a su vez enlazado al programa **Matlab**, que se encarga de procesar los datos enviados por la placa, nos muestra en tiempo real la temperatura recogida por el medidor de temperatura, crea gráficas entre valores de tiempo...

Estos datos nos van indicando la temperatura ambiente a la que se localiza. Vamos a proceder a una explicación de cómo hemos realizado el proyecto.

### Diagrama de bloques:



## Descripción del proyecto



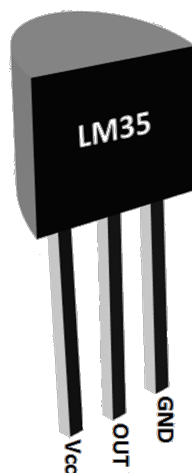
Haremos primero una pequeña descripción de [Arduino](#) MEGA (uno de los componentes más importantes utilizados en este proyecto):

[Arduino](#) es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280 y ATmega8 por su sencillez y bajo coste, que permiten el desarrollo de múltiples diseños.

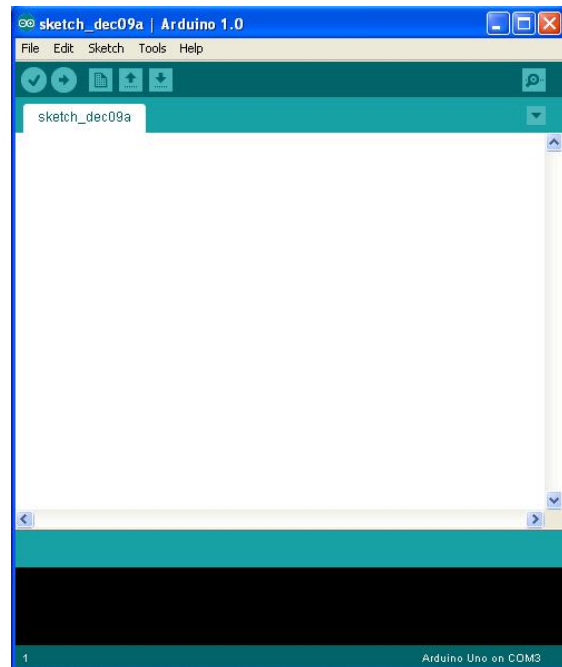
Ahora hablaremos del **sensor de temperatura LM35:**

El [LM35](#) es un sensor de temperatura con una precisión calibrada de  $\pm 1^\circ\text{C}$ . Su rango de medición abarca desde  $-55^\circ\text{C}$  hasta  $150^\circ\text{C}$ . La salida es lineal y cada grado centígrado equivale a 10mV. La tensión de salida es proporcional a la temperatura. El Conversor Análogo Digital de Arduino es de 10bits, por lo tanto para convertir el valor de lectura en Voltios usamos la siguiente expresión.  $V = (5.0 * \text{lectura}) / 1024$ . El factor adicional que aparece en la expresión es a causa de la especificación del sensor LM35, el que tiene una escala de 10mV/C por lo tanto para obtener la temperatura directamente en grados Celsius debemos multiplicar por 100.



### Programa **Arduino**:

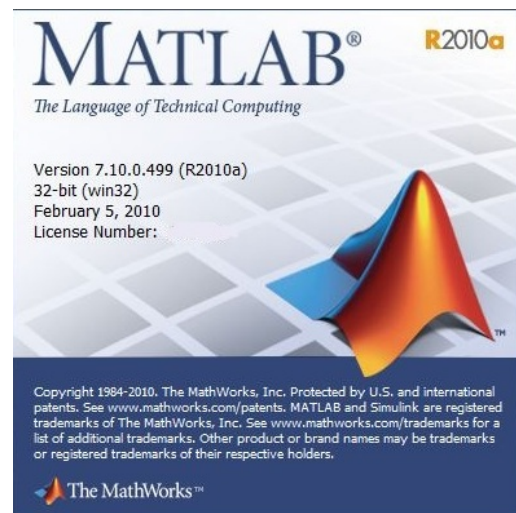
La plataforma **Arduino** se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en **Arduino**, debido a que usa la transmisión serial de datos soportada por la mayoría de los lenguajes mencionados. Para los que no soportan el formato serie de forma nativa (lenguaje de bajo nivel), es posible utilizar software intermediario que traduzca los mensajes enviados por ambas partes para permitir una comunicación fluida.



### Programa **Matlab**:

**Matlab** es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M) y servicio de especie.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware.



## Realización del proyecto

Para realizar el proyecto, primero hemos adquirido algunos conocimientos sobre **Matlab**, **Arduino** y tener el software instalado en nuestro ordenador (tanto el de **Arduino** como el de **Matlab** los podemos conseguir de sus respectivas páginas oficiales).

Primero hemos creado el programa en **Arduino**, el cual aparece a continuación:

```
// Obtencion de datos
// Declaracion de variables
float tempC;
int tempPin = 0; // Definimos la entrada en pin A0
void setup(){
  // Abre puerto serial y lo configura a 9600 bps
  Serial.begin(9600);
}
void loop(){
  // Lee el valor DESDE el sensor
  tempC = analogRead(tempPin);
  // Convierte el valor a temperatura
  tempC = (5.0 * tempC * 100.0)/1024.0;
  // Envía el dato al puerto serial
  Serial.println(tempC);
  // Tiempo cada medición
  delay(1000);
}
```

En **verde** podemos ver una pequeña descripción del código.

Después de realizar el programa hemos conectado el sensor al [Arduino](#). Para diseñar el esquema de las conexiones hemos utilizado el programa [Fritzing](#).



[Fritzing](#) es una automatización de diseño electrónico de software para diseñadores, artistas o simplemente para cualquier persona que tenga interés en la computación física y la realización de prototipos.

Su objetivo es proporcionar herramientas sencillas para documentar y compartir proyectos físicos de computación, la producción de diseños de circuitos impresos (PCB) y la electrónica de enseñanza.

En la Ilustración 1 y la Ilustración 2 vemos las conexiones realizadas.

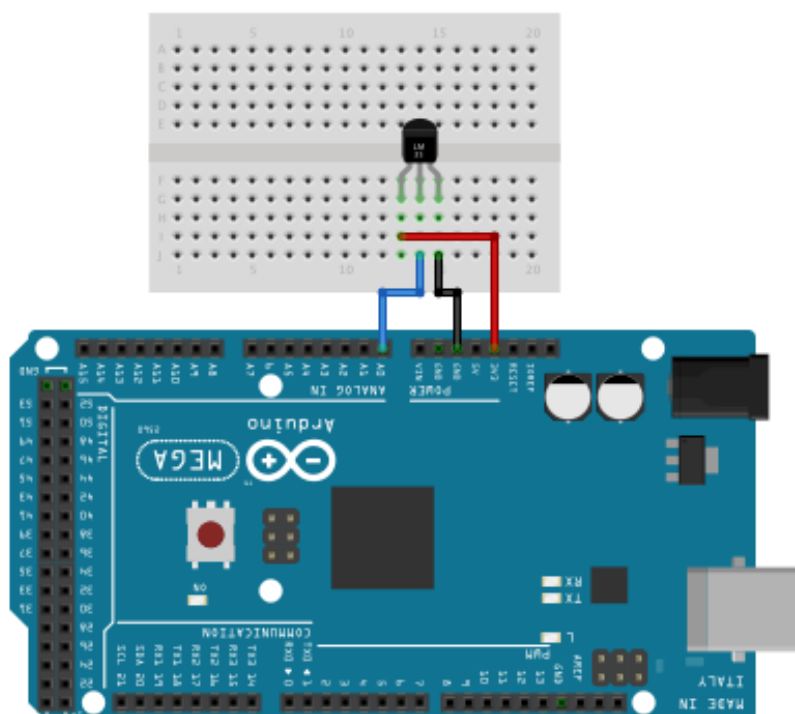
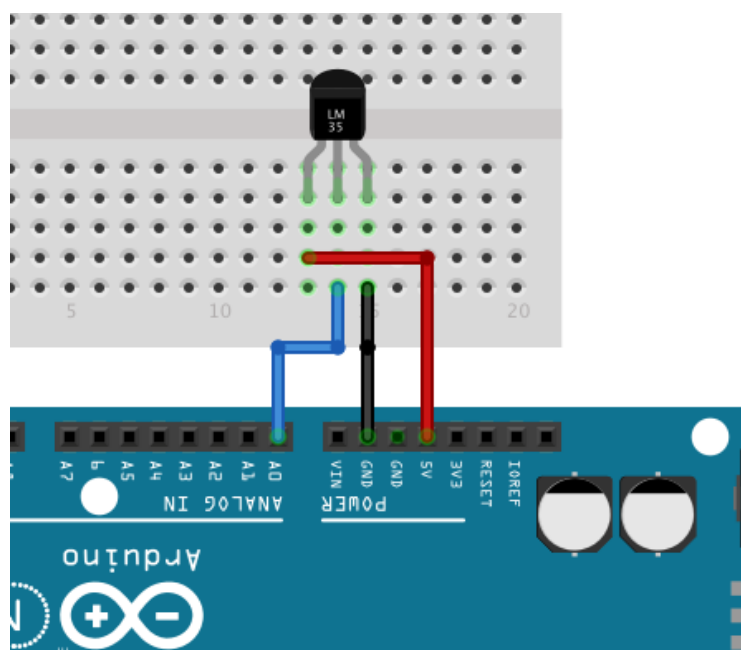


Ilustración 2

Después de esto cargamos el programa en el [Arduino](#), y en el mismo programa de [Arduino](#) vamos a Tools > serial Monitor (ilustración 3), se nos abrirá una ventana donde nos va mostrando la temperatura que recoja el sensor (ilustración 4).

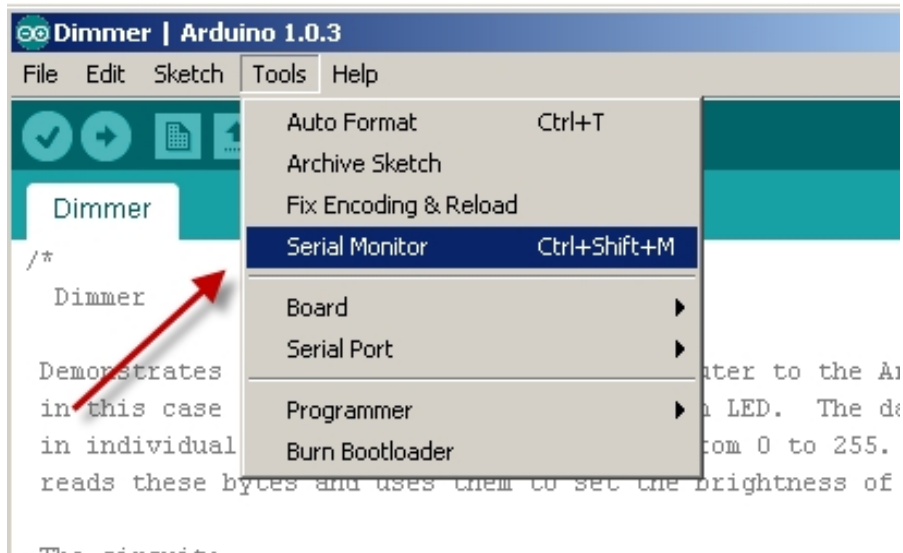


Ilustración 3

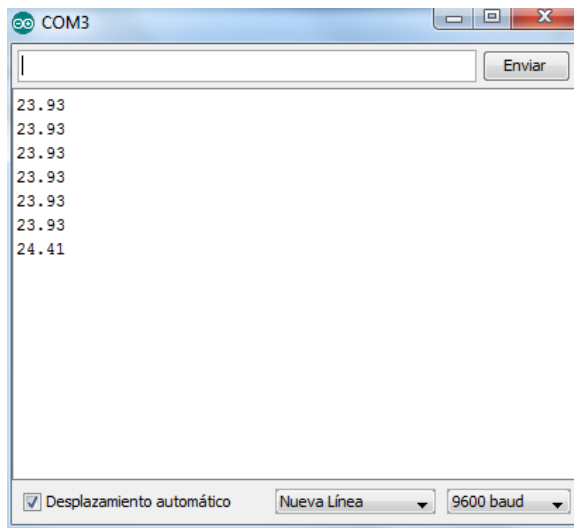


Ilustración 4

Después de probar el programa de [Arduino](#) comenzamos el programa de [Matlab](#).



Primero para conectar el [Arduino](#) al ordenador debemos conectar el cable de USB al ordenador (ilustración 5).

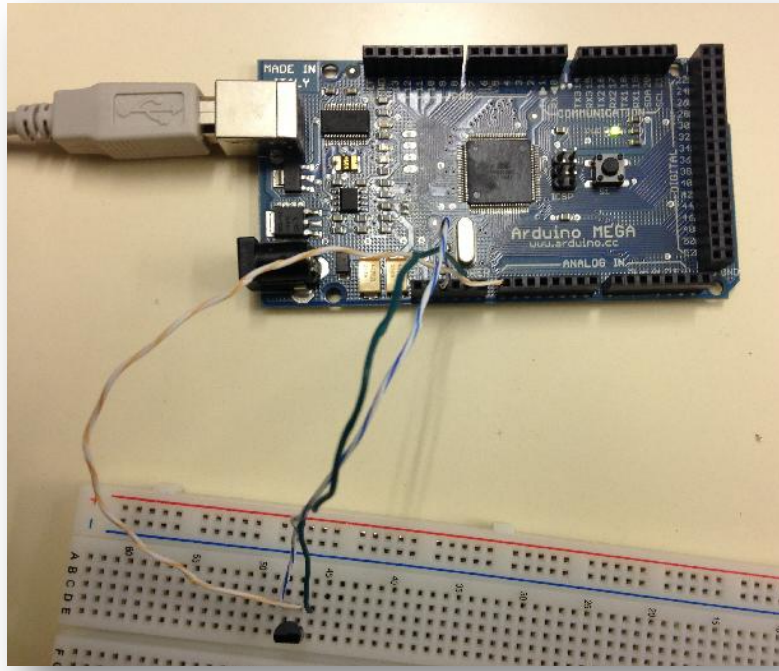


Ilustración 5

Posteriormente en el programa de [Arduino](#) debemos asignar el puerto USB conectado al ordenador para que el programa reconozca la placa.

Por último, para que el programa [Matlab](#) reconozca los datos enviados por el [Arduino](#), al principio del código de [Matlab](#) debemos asignarle de nuevo el puerto USB utilizado anteriormente.

El programa [Matlab](#) es un poco más complejo, pero buscando en ayuda de [Matlab](#) podemos encontrar todo lo deseado, de todas formas se explicará más abajo el código.

```

%-----%
%  AGUSTIN LECHUGA CARRETERO  %
%  LUIS DANIEL FIGUERO MORALES %
%-----%

%Eliminar puerto anterior
delete(instrfind({'Port'},{'COM3'}));

%Crear una conexion serie
s = serial('COM3','BaudRate',9600);
warning('off','MATLAB:serial:fscanf:unsuccessfulRead');

%Abrir el puerto
fopen(s);

%creación de la figura
scrsz = get(0,'ScreenSize');
scr1 = scrsz(3)/4;
scr2 = scrsz(4)/4;

fig = figure('Position', [scr1 scr2 2*scr1 2*scr2],...
             'Name','Termometro');
hax = axes('Units','pixels');
v = 1;

% tiempo de lectura de datos
for i = 1:10

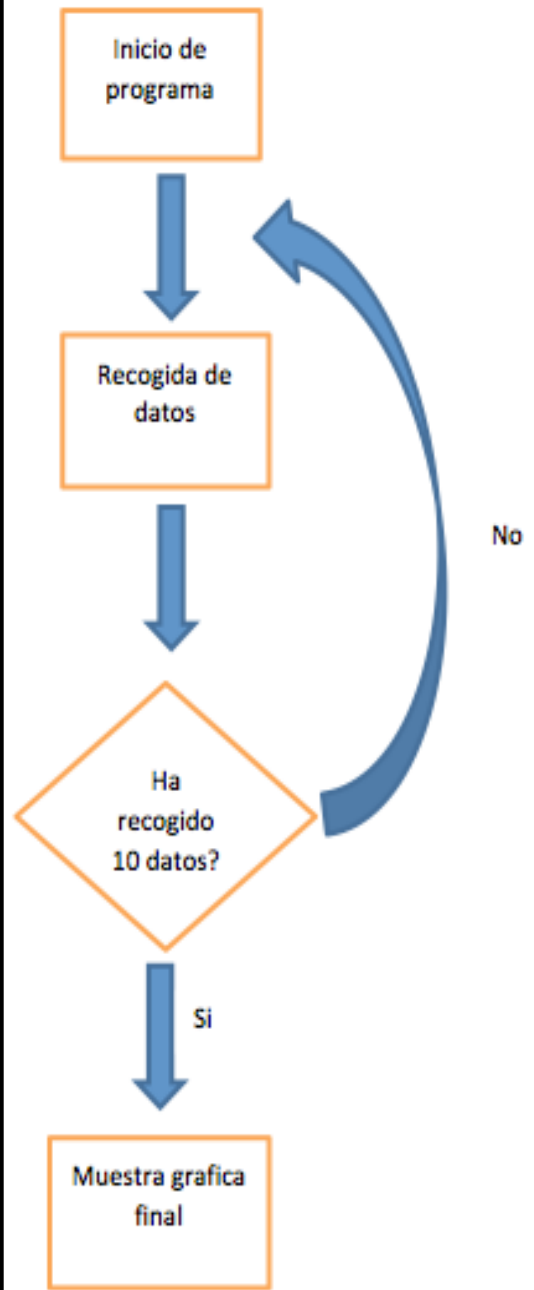
    cla

    a(i) = fscanf(s,'%f.%f');

    rectangle('Position',[0,-3,3,a(i)],'FaceColor','r')
    daspect([1,1,1])
    xlim([0,3])
    ylim([-5,100])

    pause(0.01)
end

plot(a)
    
```



En verde podemos ver una descripción del código

Al iniciar el programa en Matlab. Nos aparece una barra donde nos indica la temperatura a tiempo real (ilustración 6).

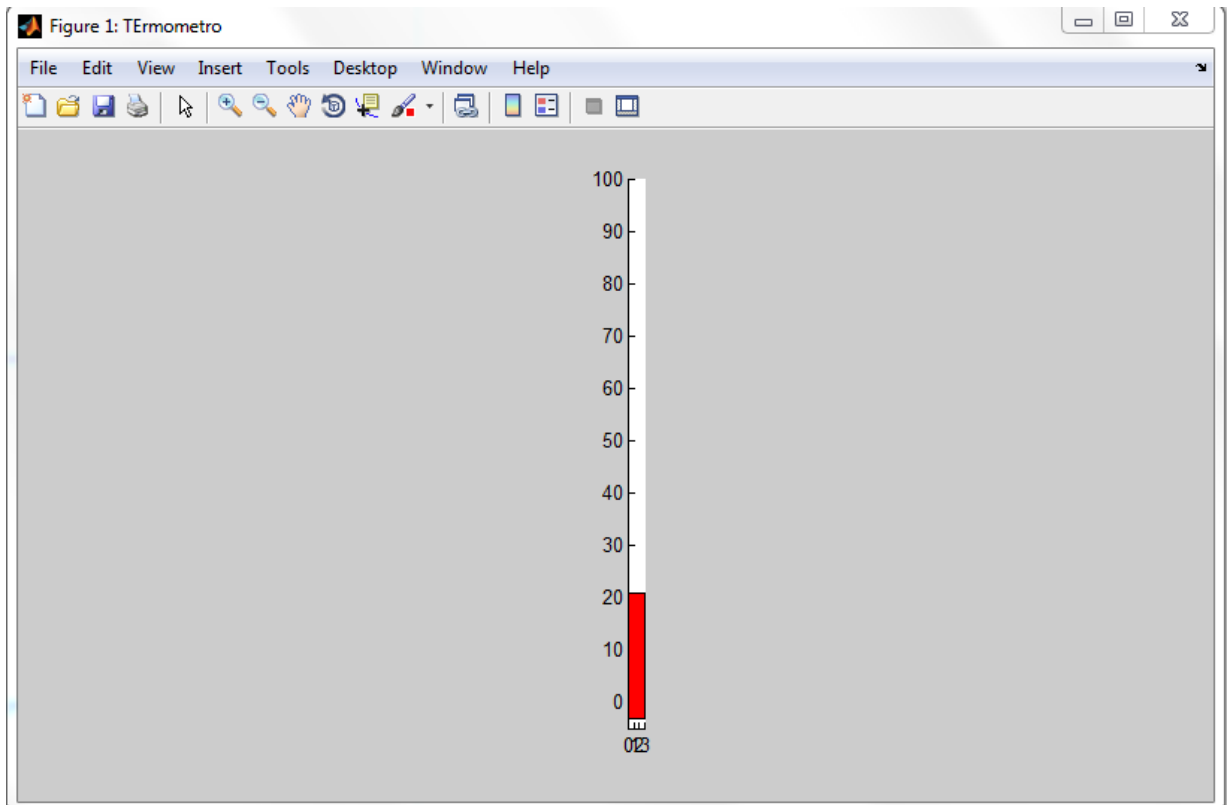


Ilustración 6

Después de ver la temperatura podemos observar una gráfica (ilustración 7) donde nos muestra la temperatura durante el periodo de tiempo que le hemos designado.

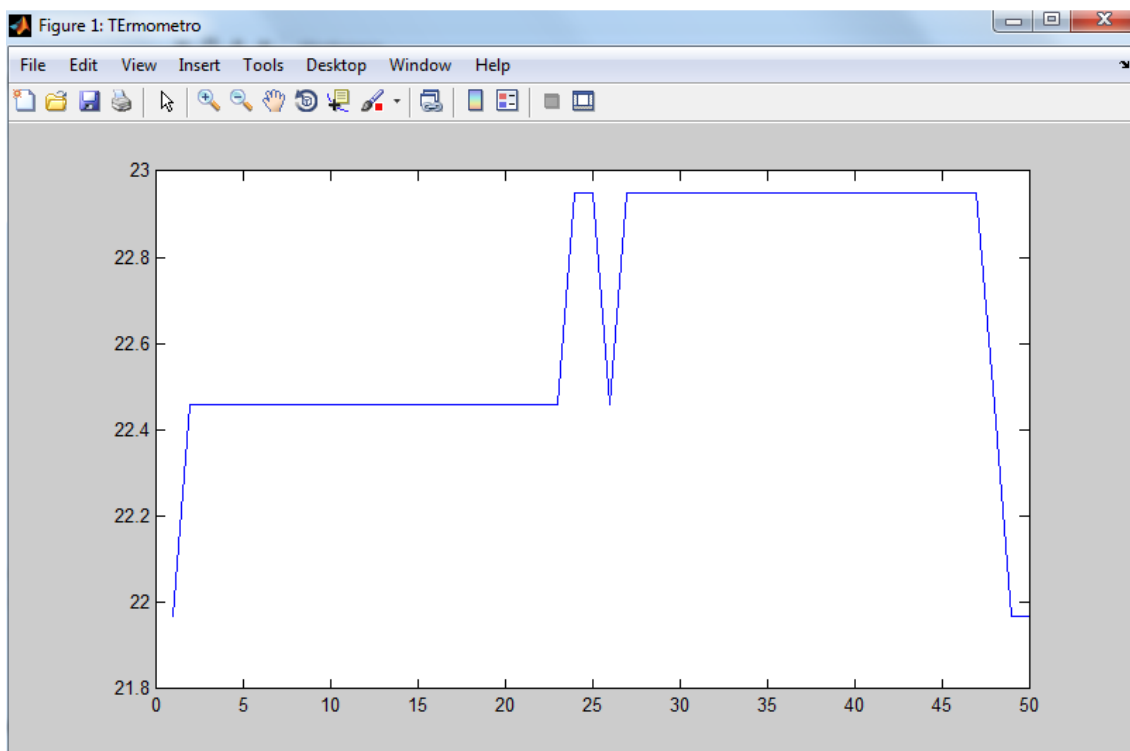


Ilustración 7

## Materiales

Los materiales utilizados para realizar el proyecto son los siguientes:

[Arduino Mega \(ATMega 1280\)](#)

Sensor de temperatura [LM35](#)

Cables conexión [protoboard](#)

[Cable USB](#) para conectar [Arduino](#)

Ordenador con software [Arduino](#) y [Matlab](#)

## Resultados y valoración

Este proyecto está pensado para que cualquier persona con un bajo nivel en programación y montaje de circuitos pueda iniciarse en este tipo de montajes y poder obtener resultados de medición de temperatura y así dar pie a introducirse en la creación de nuevos montajes más sofisticados para obtener los datos de otras variables del entorno.

## Bibliografía y fuentes

<https://es.wikipedia.org/>

<http://www.arduino.cc/>

<http://es.mathworks.com/>

<http://fritzing.org/>