



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE

PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

CONTROL DE PROCESOS POR COMPUTADORA

EDEN PEREZ ROMERO 200831000

CONTROL III

ROBERTO LAURO HERNANDEZ RIVERA 200421340

PROFESOR: DR. JAIME JULIAN CID MONJARAZ

REPORTE PRÁCTICA 2

Facultad de Ciencias
de la Electrónica
PRIMAVERA 2013

Practica 2

ARDUINO Y SENSOR DE TEMPERATURA LM 35

Objetivos:

Realizar la interfaz para poder graficar la señal procedente de un sensor de temperatura, en este caso en particular de un sensor lm35 y así poder visualizar en una pantalla LCD los valores de temperatura y también observarlos en el monitor del puerto serial del software de arduino. Con los datos obtenidos poder saber cómo es la función de transferencia de la señal del sensor lm35.

Marco Teórico

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos con la que ya dimos una introducción en la práctica anterior, así que hablaremos de los componentes como el sensor lm35

LM35

El LM35 es un sensor de temperatura con una precisión calibrada de 1°C y un rango que abarca desde -55° a +150°C.

El sensor se presenta en diferentes encapsulados pero el más común es el to-92 de igual forma que un típico transistor con 3 patas, dos de ellas para alimentarlo y la tercera nos entrega un valor de tensión proporcional a la temperatura medida por el dispositivo. Con el LM35 sobre la mesa las patillas hacia nosotros y las letras del encapsulado hacia arriba tenemos que de izquierda a derecha los pines son: VCC – V out - GND.

La salida es lineal y equivale a 10mV/°C por lo tanto:

$$+1500\text{mV} = 150^{\circ}\text{C}$$

$$+250\text{mV} = 25^{\circ}\text{C}$$

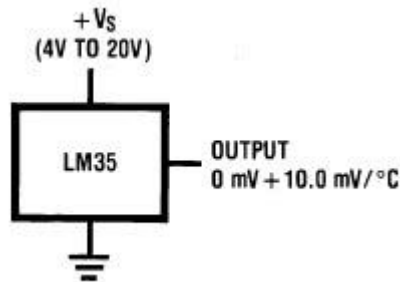
$$-550\text{mV} = -55^{\circ}\text{C}$$

Funcionamiento: Para hacernos un termómetro lo único que necesitamos es un voltímetro bien calibrado y en la escala correcta para que nos muestre el voltaje equivalente a temperatura. El LM35 funciona en el rango de alimentación comprendido entre 4 y 30 voltios.

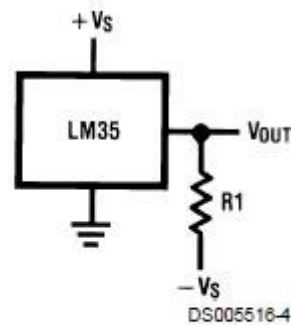
Podemos conectarlo a un conversor Analógico/Digital y tratar la medida digitalmente, almacenarla o procesarla con un μ Controlador o similar.

Usos: El sensor de temperatura puede usarse para compensar un dispositivo de medida sensible a la temperatura ambiente, refrigerar partes delicadas del robot o bien para loggear temperaturas en el transcurso de un trayecto de exploración.

1. Sensor de temperatura básico (+2 °C a 150 °C):

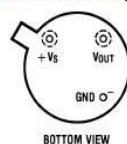


2. Sensor de temperatura con rango completo:



Choose $R_1 = -V_S / 50 \mu A$
 $V_{OUT} = +1,500 \text{ mV at } +150^\circ\text{C}$
 $= +250 \text{ mV at } +25^\circ\text{C}$
 $= -550 \text{ mV at } -55^\circ\text{C}$

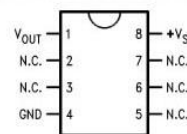
TO-46
Metal Can Package*



BOTTOM VIEW

*Case is connected to negative pin (GND)
 LM35H, LM35AH, LM35CH, LM35CAH or
 LM35DH
 See NS Package Number H03H

SO-8
Small Outline Molded Package



N.C. = No Connection

Top View
 Order Number LM35DM
 See NS Package Number M08A

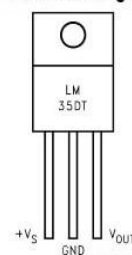
TO-92
Plastic Package



BOTTOM VIEW
 DS005516-2

Order Number LM35CZ,
 LM35CAZ or LM35DZ
 See NS Package Number Z03A

TO-220
Plastic Package*



*Tab is connected to the negative pin (GND).
 Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
 See NS Package Number TA03F

LCD 16 X 2

Descripción general:

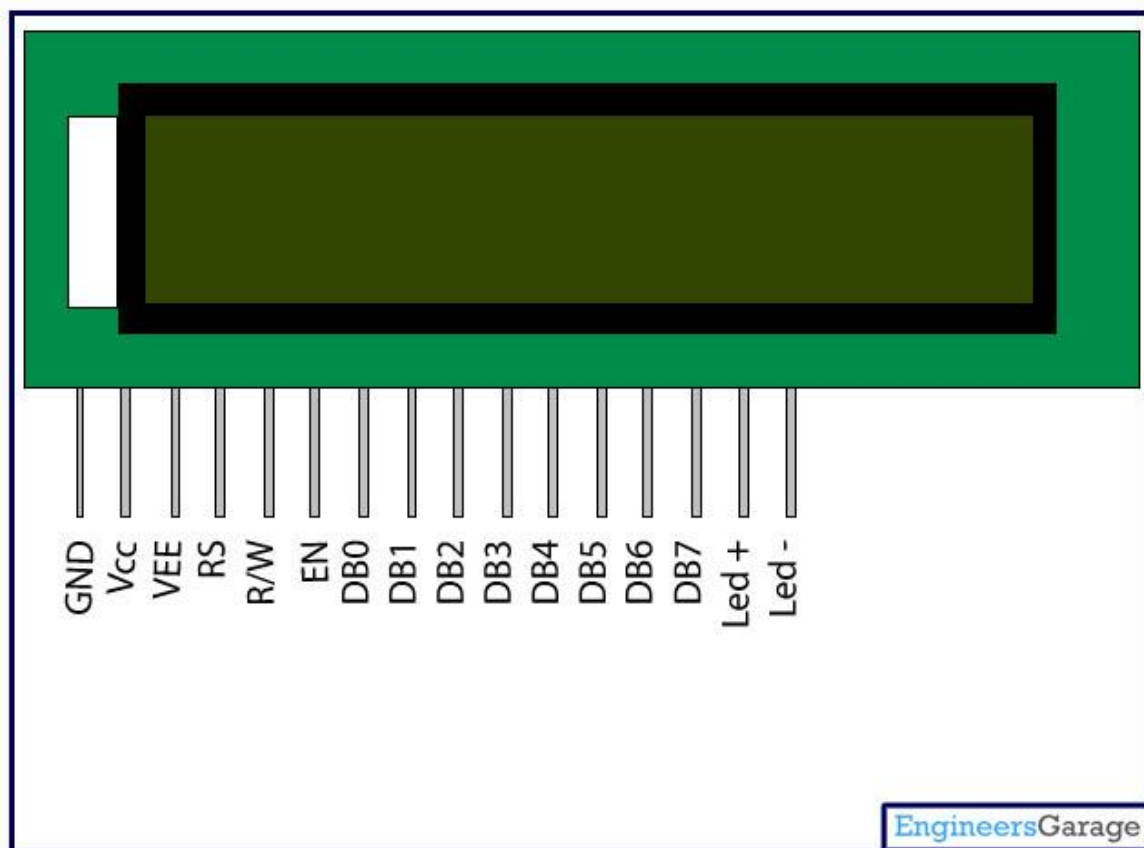
El LCD es actualmente el circuito más barato y confiable para mostrar datos en un proceso de monitoreo y control. Su interfaz con los controladores se realiza a través de un conector de 14 pines.

Todos los fabricantes del display de cristal líquido (“Liquid Cristal Display”) LCD, han estandarizado sus señales en el conector de 14 pines, así como sus comandos de control para el manejo del mismo

En el LCD se pueden mostrar datos como la hora y la fecha, así como valores de variables tales como nivel, presión, gasto, temperatura, etc.

El LCD puede también emplearse para mostrar parámetros internos del sistema, de acuerdo a su aplicación o para mostrar al usuario las opciones de configuración mientras lo opera.

El módulo LCD lleva integrado a sus circuitos una memoria ROM conocida como “generador de caracteres” que habrá de generar los patrones de la matriz de puntos (5 x 7 ó 7 x 9) que forman los caracteres en la pantalla. También tiene una RAM interna que almacena los caracteres y los exhibe en el módulo LCD.



Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	V _{cc}
3	Contrast adjustment; through a variable resistor	V _{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V _{cc} (5V)	Led+
16	Backlight Ground (0V)	Led-

MATERIAL:

ARDUINO MEGA 2560

PROTOBOARD

PANTALLA LCD 16 X 2

LM 35

DESARROLLO:

La programación para el arduino es la siguiente, esta comentado para saber para qué sirve cada línea de código pero en general necesitamos incluir o declarar la librería de la pantalla de cristal líquido, después declaramos las entradas que utilizaremos para la pantalla LCD. Y también mostramos los datos en el monitor serial.

```
#include <LiquidCrystal.h> //Incluir esta libreria para poder usar el lcd
```

```
int Ana1 = A0; //Entrada analogica de LM35
LiquidCrystal lcd(12,11,5,4,3,2); //Definimos la pantalla LCD
int Temp = 0;
char Grados = '0';
```

```
void setup(){
  Serial.begin(9600);
  lcd.begin(16,2);
  pinMode(13,OUTPUT);
  digitalWrite(13, HIGH); //Activamos la retroiluminacion
}
```

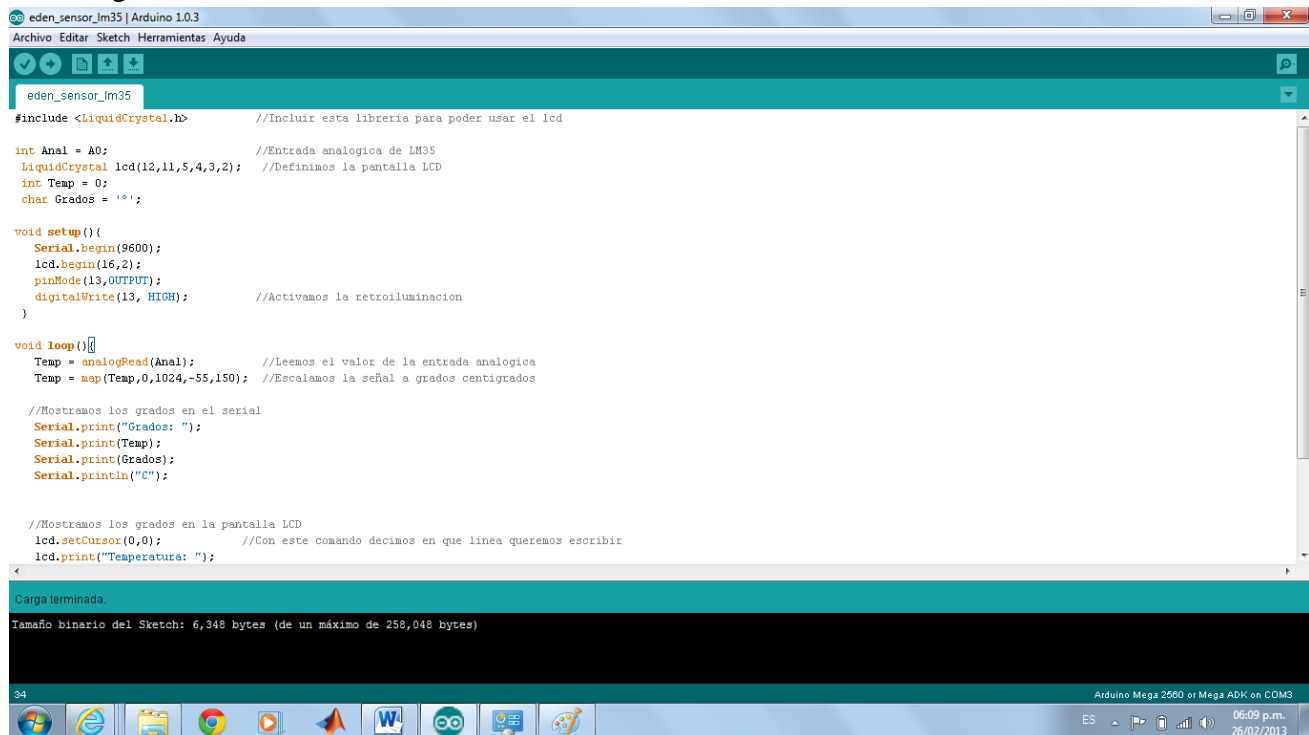
```
void loop(){
  Temp = analogRead(Ana1); //Leemos el valor de la entrada analogica
  Temp = map(Temp,0,1024,-55,150); //Escalamos la señal a grados centígrados

  //Mostramos los grados en el serial
  Serial.print("Grados: ");
  Serial.print(Temp);
  Serial.print(Grados);
  Serial.println("C");

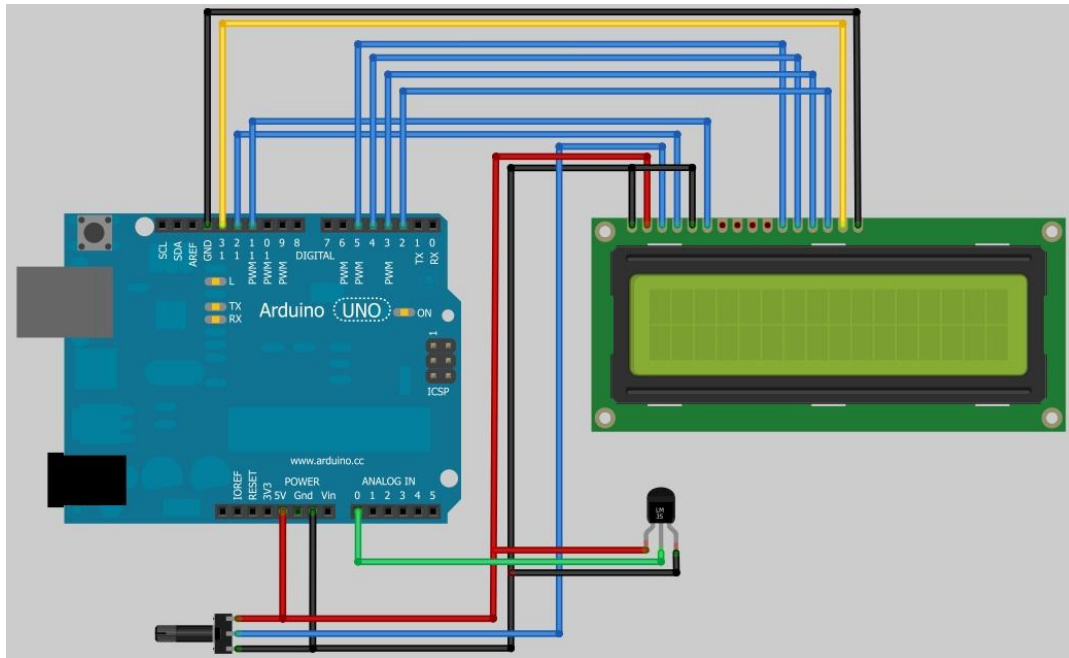
  //Mostramos los grados en la pantalla LCD
  lcd.setCursor(0,0); //Con este comando decimos en que linea queremos escribir
  lcd.print("Temperatura: ");
  lcd.setCursor(0,1);
  lcd.print(Temp);

  delay(10000); //Al ser temperatura no hace falta leerlo tan seguido
}
```

Y lo cargamos al arduino

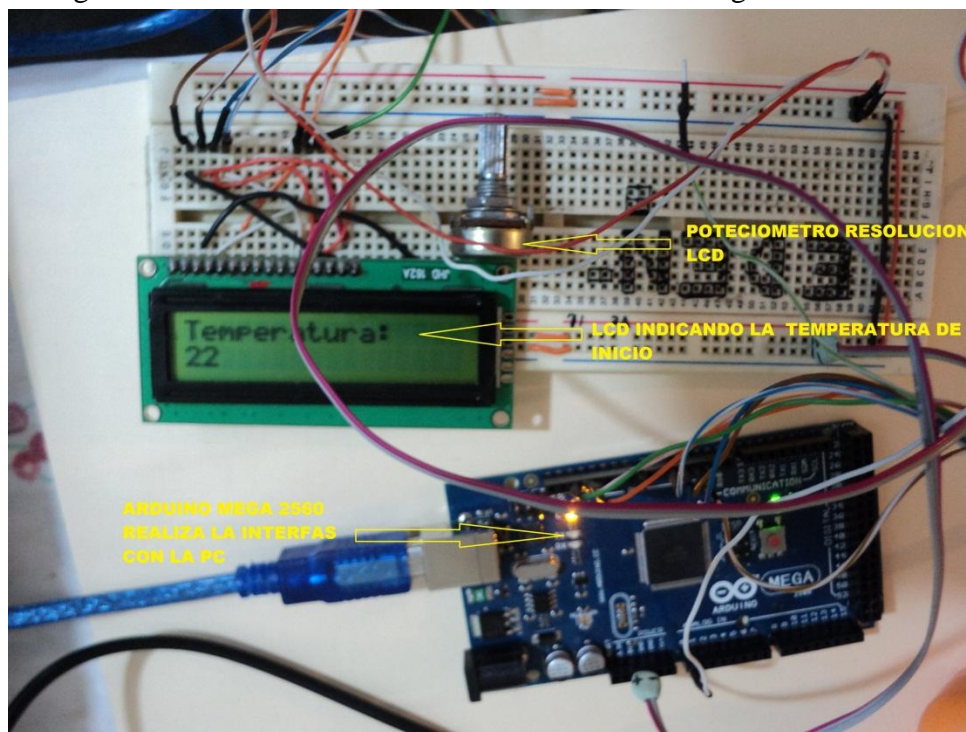


Ahora necesitamos armar el siguiente circuito en un protoboard

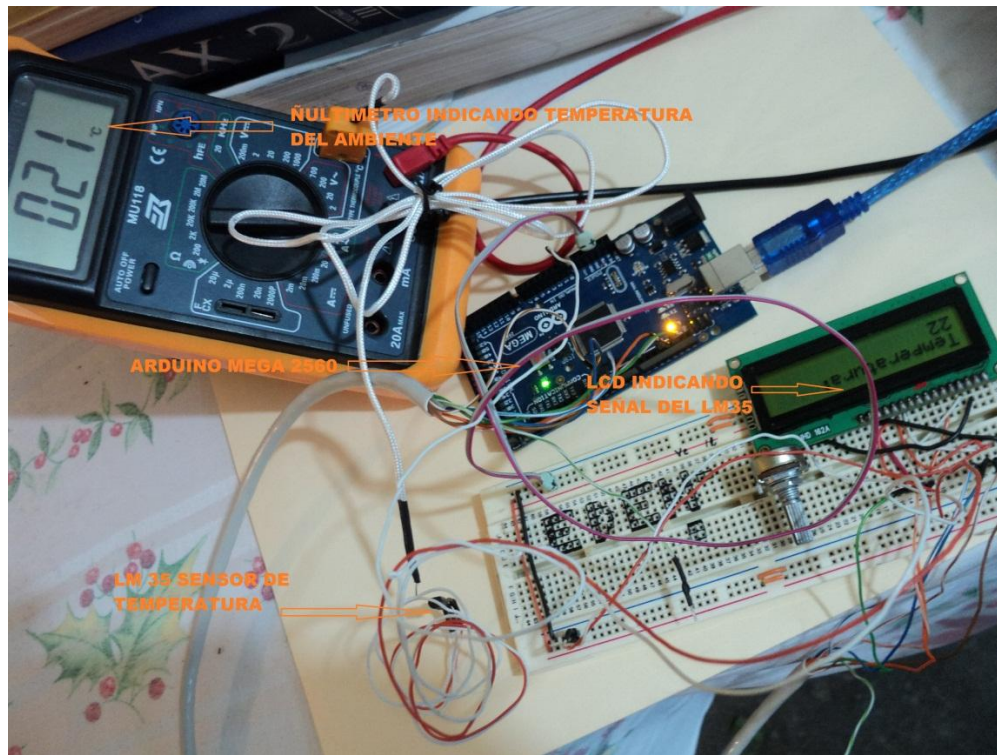


Después obtenemos los datos tanto en el LCD como en el monitor serial del software de arduino como se muestra en las siguientes imágenes.

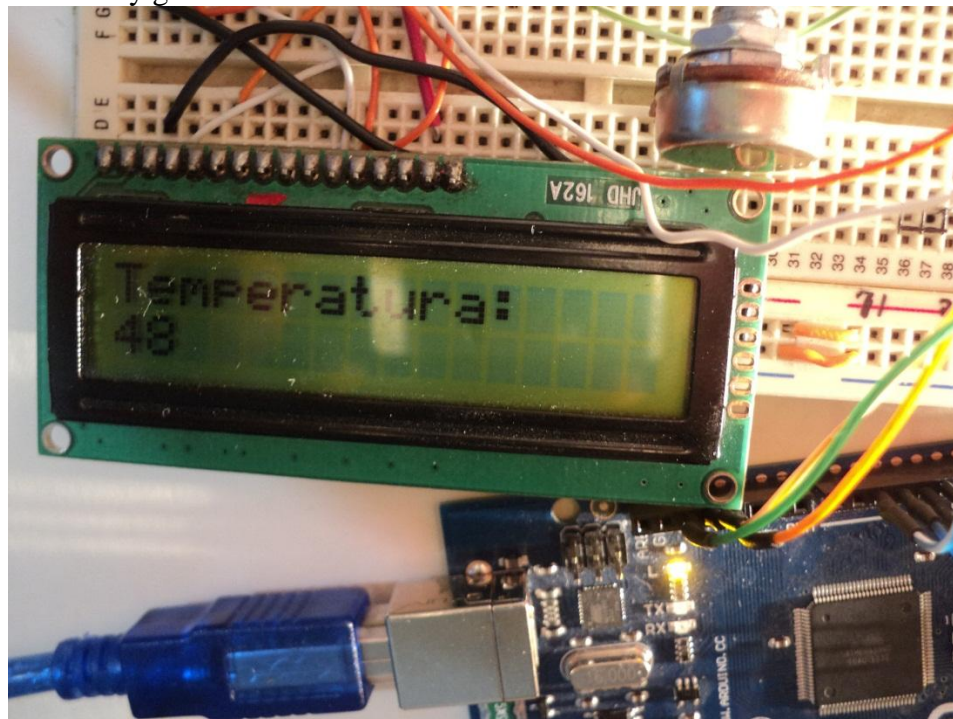
Obtenemos los datos que nos proporciona el sensor lm35 a temperatura ambiente y el sensor configurado en su forma básica como se muestra en la siguiente



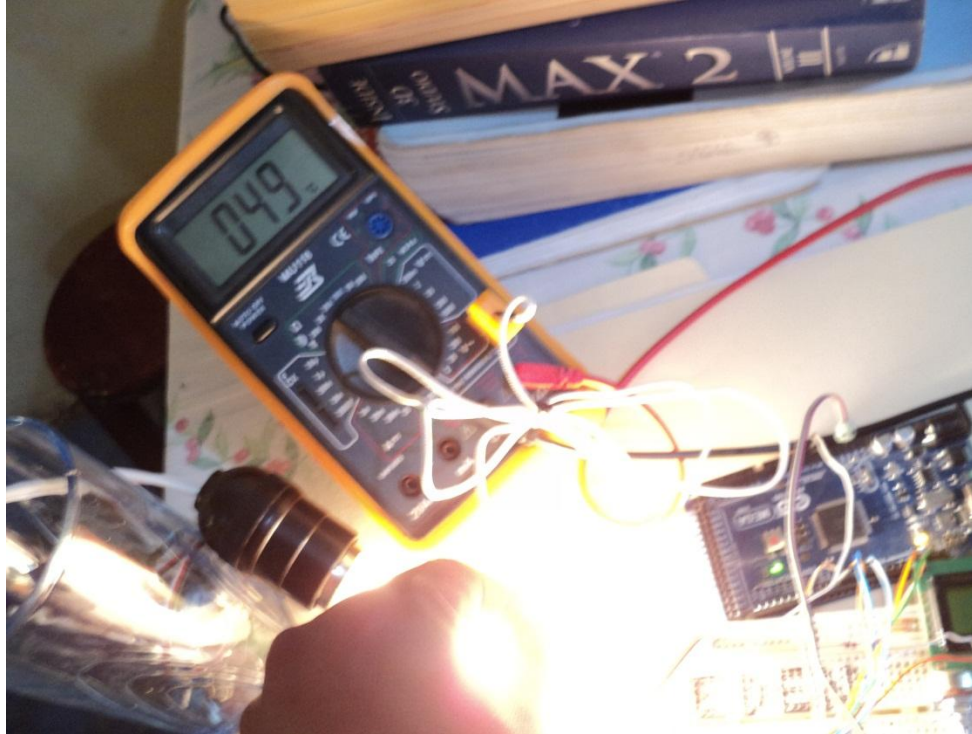
Utilizando un multímetro con medidor de temperatura compramos que la señal del sensor lm 35 nos indique la temperatura que está en el ambiente



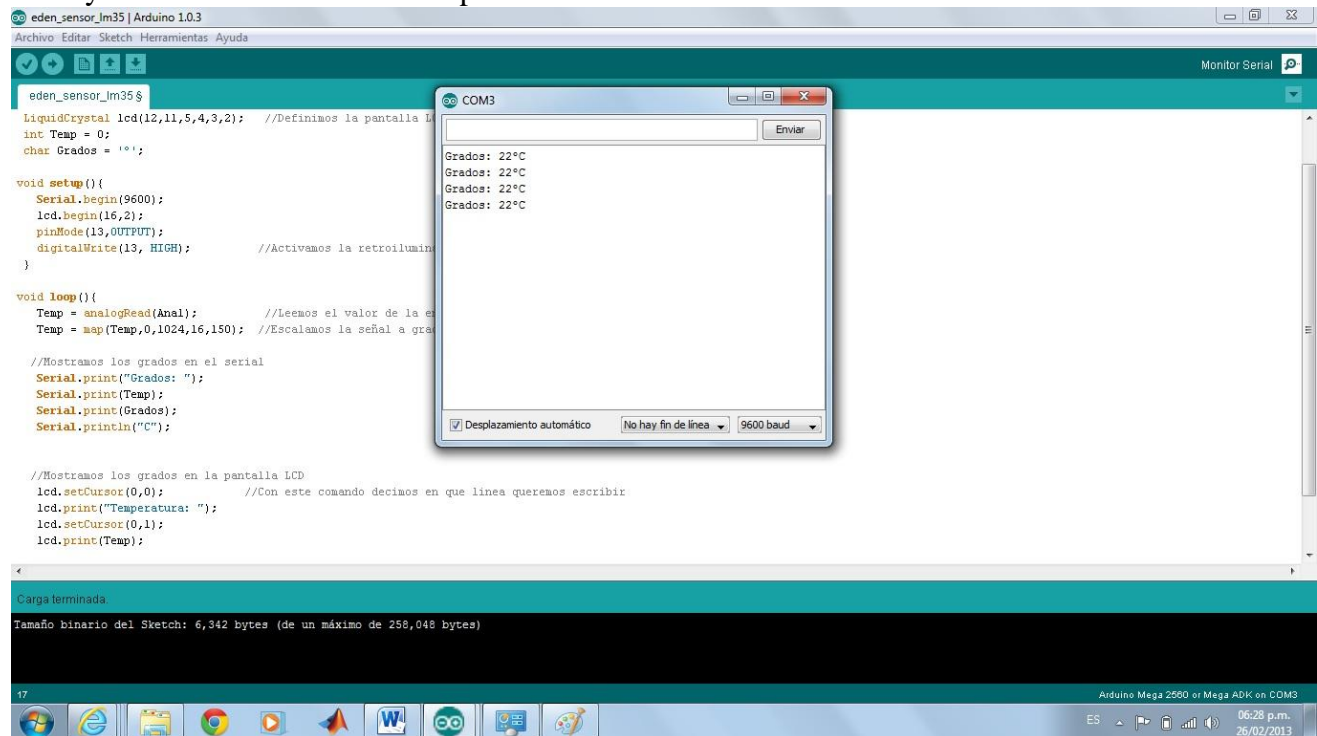
Después con ayuda de un foco calefactor aumentamos la temperatura del sensor y observamos dichos datos obtenidos con el sensor mediante la LCD y en el monitor del puerto serial del software de arduino y lo verificamos con ayuda del multímetro para posteriormente seguir con la obtención de los datos y graficarlos en matlab



En la siguiente imagen verificamos la medición de temperatura con el multímetro



La siguiente imagen nos muestra la lectura de la temperatura del medio ambiente censada con el lm35 y mostrada en el monitor del puerto serial del mismo software del arduino



En la imagen siguiente es la recopilación de datos al poner el foco calefactor en el lm35 y así obtener los datos necesarios del comportamiento del sensor lm35

eden_sensor_lm35 | Arduino 1.0.3

Archivo Editar Sketch Herramientas Ayuda

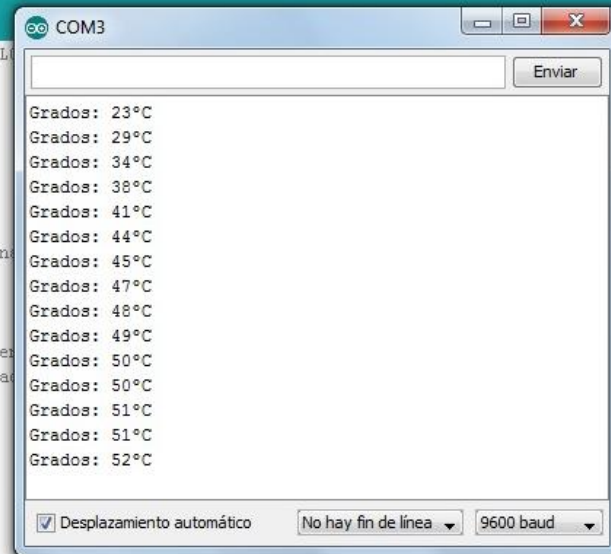
eden_sensor_lm35 \$

```
liquidCrystal lcd(12,11,5,4,3,2); //Definimos la pantalla LCD
int Temp = 0;
char Grados = '°';

void setup(){
  Serial.begin(9600);
  lcd.begin(16,2);
  pinMode(13,OUTPUT);
  digitalWrite(13, HIGH); //Activamos la retroiluminación

  //Mostramos los grados en el serial
  Serial.print("Grados: ");
  Serial.print(Temp);
  Serial.print(Grados);
  Serial.println("C");

  //Mostramos los grados en la pantalla LCD
  lcd.setCursor(0,0); //Con este comando decimos en que linea queremos escribir
  lcd.print("Temperatura: ");
```



GRAFICAR LOS DATOS OBTENIDOS EN MATLAB

Para la siguiente parte de la práctica hicimos un programa en matlab el cual nos permitiera recuperar los datos obtenidos con la interfaz de arduino y la pc

El siguiente código es una función hecha en matlab para obtener los datos del lm35 y graficarlos en matlab, y lo hacemos con la lectura del puerto serial, donde la función la nombramos Matlab_arduino y la mandamos a llamar con este nombre el cual nos pregunta por el numero de muestras que deseamos que lea en el puerto serial.

```
function Matlab_Arduino(numero_muestras)

% Matlab + Arduino Serial Port communication

close all;
clc;
y=zeros(1,1000); %Vector donde se guardarán los datos

%Inicializo el puerto serial que utilizaré
delete(instrfind({'Port'}, {'COM3'}));
puerto_serial=serial('COM3');
puerto_serial.BaudRate=9600;
warning('off', 'MATLAB:serial:fscanf:unsuccessfulRead');

%Abro el puerto serial
%%fopen(Port_0002.Hub_0001);
fopen(puerto_serial);
```

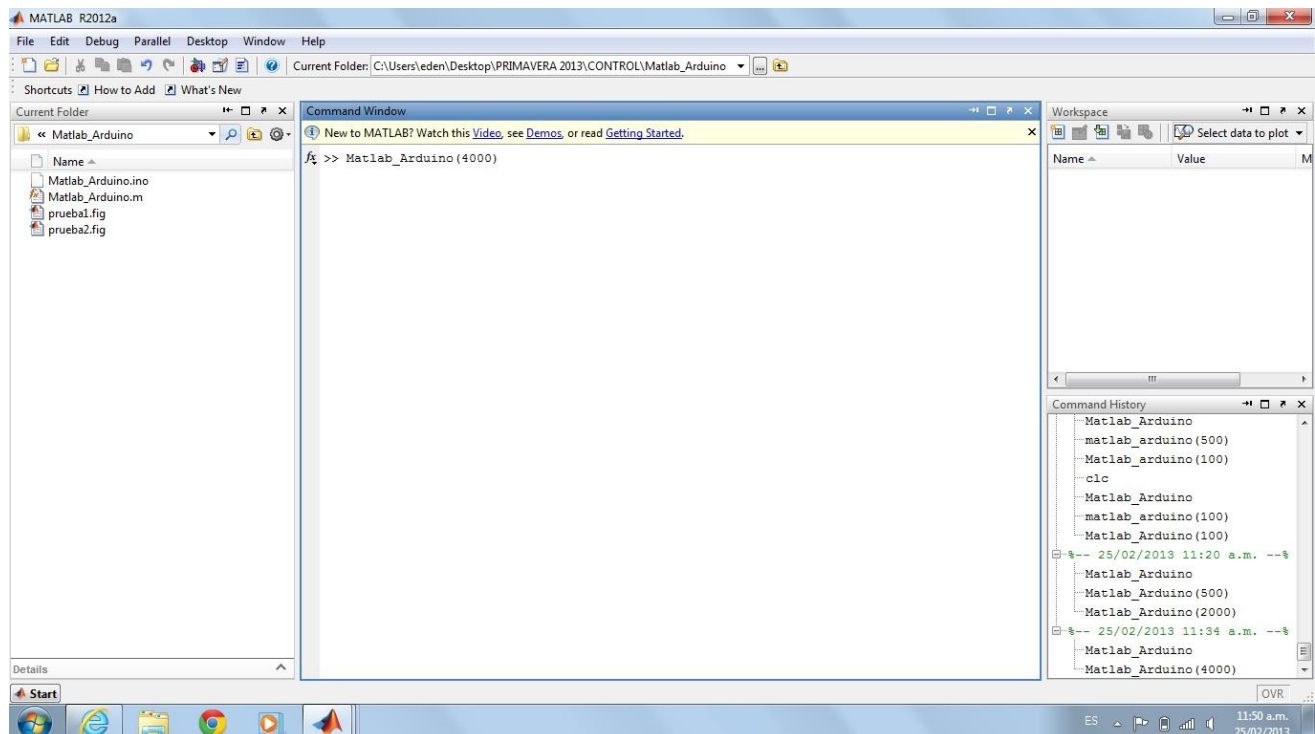
```
%Declaro un contador del número de muestras ya tomadas
contador_muestras=1;

%Creo una ventana para la gráfica
figure('Name','arduino-matlab')
title('SERIAL COMMUNICATION MATLAB+ARDUINO');
xlabel('Número de muestra');
ylabel('Temp (T)');
grid on;
hold on;

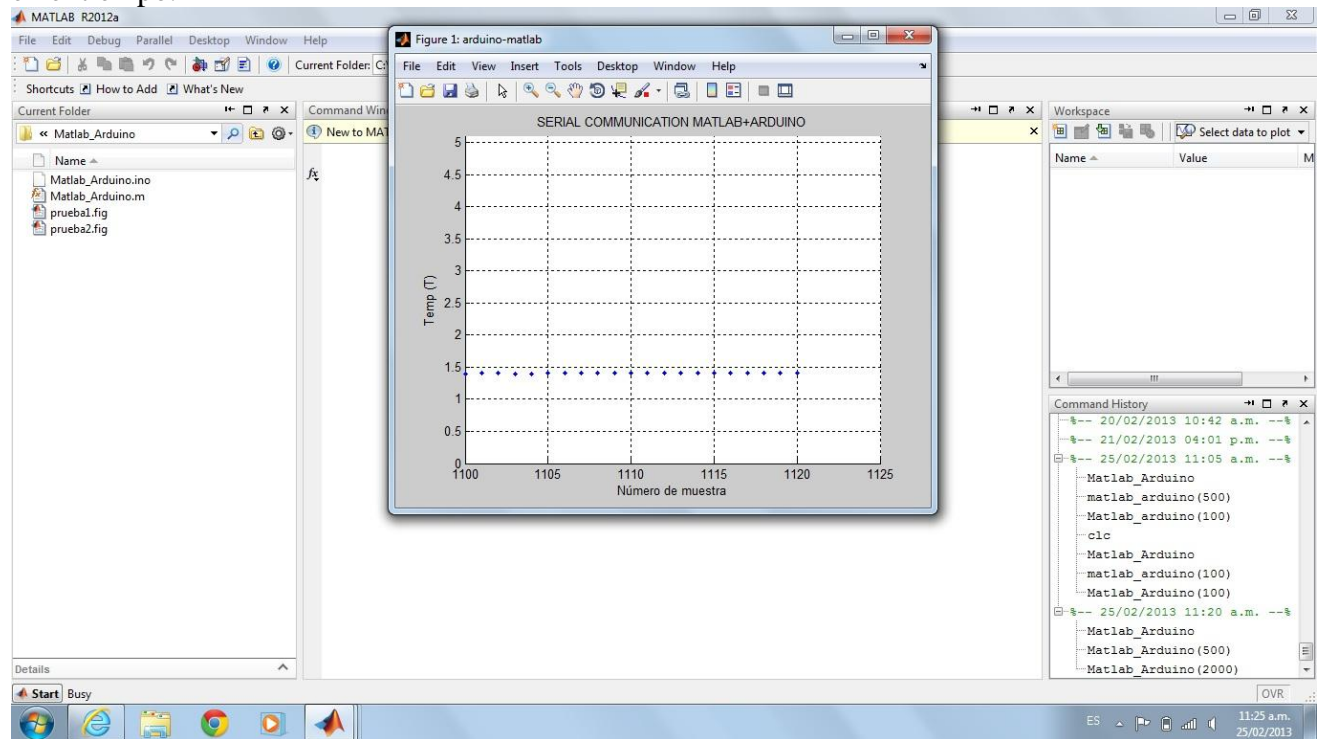
%Bucle while para que tome y dibuje las muestras que queremos
while contador_muestras<=numero_muestras
    ylim([0 5.1]);
    xlim([contador_muestras-20 contador_muestras+5]);
    valor_potenciometro=fscanf(puerto_serial,'%d');
    y(contador_muestras)=(valor_potenciometro(1))*5/1024;
    plot(contador_muestras,y(contador_muestras),'.');    %%X-r
    drawnow
    contador_muestras=contador_muestras+1;
end

%Cierro la conexión con el puerto serial y elimino las variables
fclose(puerto_serial);
delete(puerto_serial);
clear all;

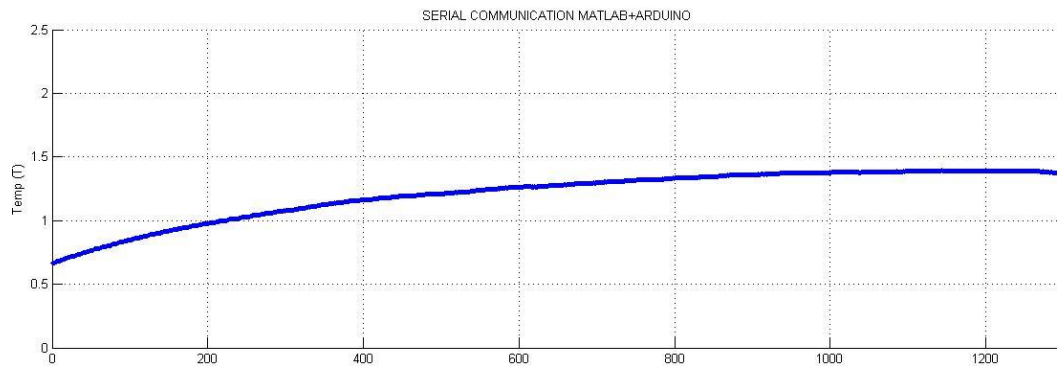
end
```

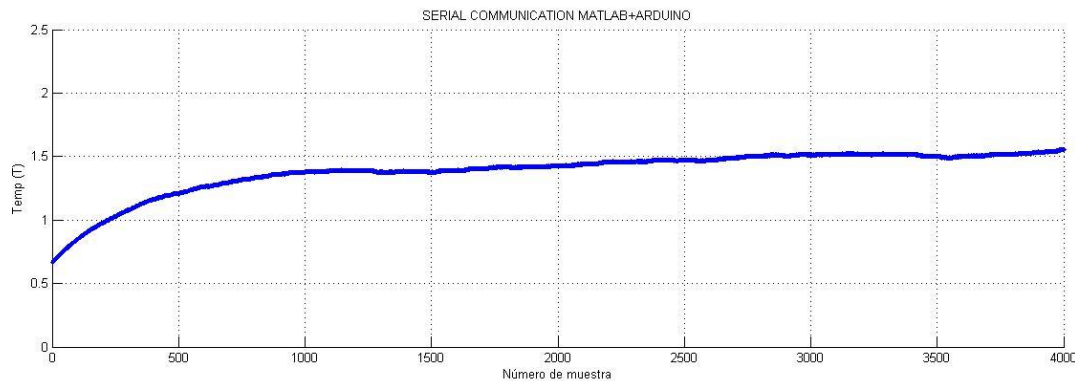


Después de que introducimos el número de muestras entonces comienza a obtener las muestras y a graficarlas cada en nuestro caso obtuvimos 4000 muestras para poder visualizar el cambio del voltaje en el tiempo.



Y cambiando la escala del eje X y Y para ver mejor la gráfica de los datos obtenidos nos quedo la siguiente grafica





Por lo tanto la gráfica que obtuvimos nos indica que nuestra función de transferencia tiene con una respuesta transitoria tipo escalón

También para poder obtener la lectura con este programa en matlab fue necesario realizar un programa para el arduino y así poder leer los datos que se estaban obteniendo en la entrada analógica A0 del arduino mega 2560 el cual solo indica que debe leer la entrada analógica A0 y poder leerlo a través del puerto serial el cual ya está declarado en nuestra función hecha en matlab

```
int potenciometro_pin=0; //Analog 0
int valor_potenciometro=0;

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

void loop(){
  valor_potenciometro=analogRead(potenciometro_pin);
  Serial.println(valor_potenciometro);
  delay(100);
}
```

Conclusiones

La importancia de esta práctica es saber obtener los datos de la respuesta de un sensor nos sirve para poder caracterizarlo y así poder tener la respuesta de nuestro sistema a controlar y con la gráfica poder realizar los pasos para hacer el controlador que necesitamos para que nuestra planta funcione como se desea y con ayuda de la herramienta como lo es el software de matlab que nos puede facilitar el proceso en la realización del controlador.



Referencias

<http://wechoosethemoon.es/2011/07/15/arduino-matlab-adquisicion-de-datos/>

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

<http://arduino.cc/en/Guide/Windows>

<http://playingwitharduino.blogspot.mx/2012/01/lm35-temperatura-y-lcd.html>