Benemérita Universidad

Autónoma de Puebla

Control de Procesos Por Computadora

Practica II

Caracterización de una incubadora

Catedrático:

Dr. Jaime Julián Cid Monjaraz

Alumnos:

- Estanislao Sierra Jair Ernesto (200815864)
- Sanzón Parra José Manuel (200816420)
- Peña Sánchez Emmanuel(2008)

Puebla, Pue. A 8 de Febrero de 2013

Objetivo

Realizar la lectura de un sensor LM35 mediante la interfaz entre la Tarjeta de Desarrollo Arduino y el software Matlab, para obtener su grafica y posteriormente realizar la función de transferencia que describe al sensor.

Marco Teórico

La función de transferencia es la forma básica de describir modelos de sistemas lineales. Basada en la transformación de Laplace, permite obtener la respuesta temporal, la respuesta estática y la respuesta en frecuencia. El análisis de distintas descomposiciones de la respuesta temporal permite adquirir útiles ideas cualitativas, y definir varios importantes conceptos: efectos de las condiciones iniciales, respuesta libre y forzada, regímenes permanente y transitorio. También permite definir el concepto central de estabilidad y establecer un primer criterio para su investigación.

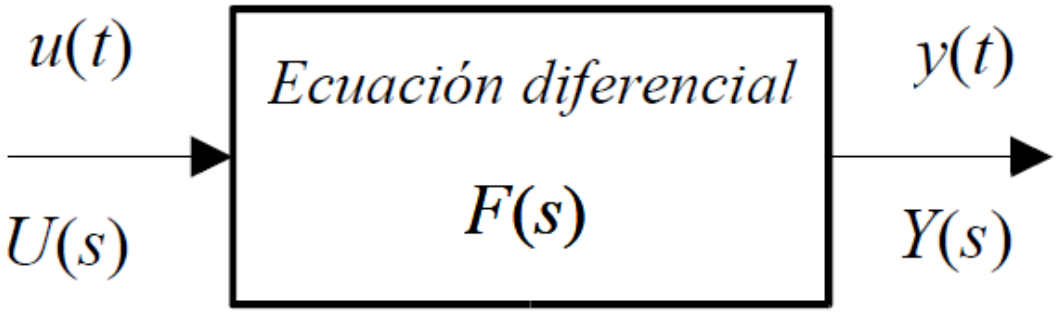


Diagrama de Bloques Básico

$$Y(s) = U(s) \cdot F(s) \qquad F(s) = \frac{B(s)}{A(s)}$$

La salida puede expresarse (en transformadas) como la entrada multiplicada por la función de transferencia F(s) del sistema, expresada como cociente de polinomios.

Desarrollo

Para poder realizar esta practica primero se tuvo que bajar un software (ArduinoIO) que se descargo de la pagina de matlab y se instalo en el mismo. De igual forma también se cargo un programa en el arduino que permitiera que matlab pudiese reconocer el puerto de comunicación en el cual esta conectado en arduino.

Mediante las siguientes instrucciones se realizaron pruebas con el Arduino.

Connect

- Use the command `a=arduino('port')`, with the right COM port as a string input argument, to connect MATLAB with the board and create an arduino object in the workspace:

```
>> a=arduino('COM5');
```

Assign Pin Mode (input/output)

- Use the command `a.pinMode(pin,str)` to get or set the mode of a specified pin:

- Examples:

```
>> a.pinMode(11,'output')
>> a.pinMode(10,'input')
>> val=a.pinMode(10)
>> a.pinMode(5)
>> a.pinMode;
```

Digital Read (digital input)

- Use the command `a.digitalRead(pin)` to read the digital status of a pin:

  Examples:
  ```
  >> val=a.digitalRead(4)
  ```

  This returns the value (0 or 1) of the digital pin number 4

Digital Write (digital output)

- Use the command `a.digitalWrite(pin,val)` with the pin as first argument and the value (0 or 1) as second argument:

- Examples:
  ```
  a.digitalWrite(13,1); % sets pin #13 high
  a.digitalWrite(13,0); % sets pin #13 low
  ```

Analog Read (analog input)

- Use the command `val=a.analogRead(pin)` with the pin as an integer argument:

- Example:
  ```
  val=a.analogRead(0); % reads analog pin # 0
  ```

- The returned argument ranges from 0 to 1023

- Note that 6 analog input pins (0 to 5) coincide with the digital pins 14 to 19 and are located on the bottom right corner of the board

Analog Write (analog output)

- Use the command `a.analogWrite(pin,val)` with the pin as first argument and the value (0 to 255) as second argument:

- Examples:

```
a.analogWrite(11,90); % sets pin #11 to 90
a.analogWrite(3,10); % sets pin #3 to 10
```

Disconnect

- Use the command `delete(a)` to disconnect the MATLAB session from the Arduino board:

```
>> delete(a);
```

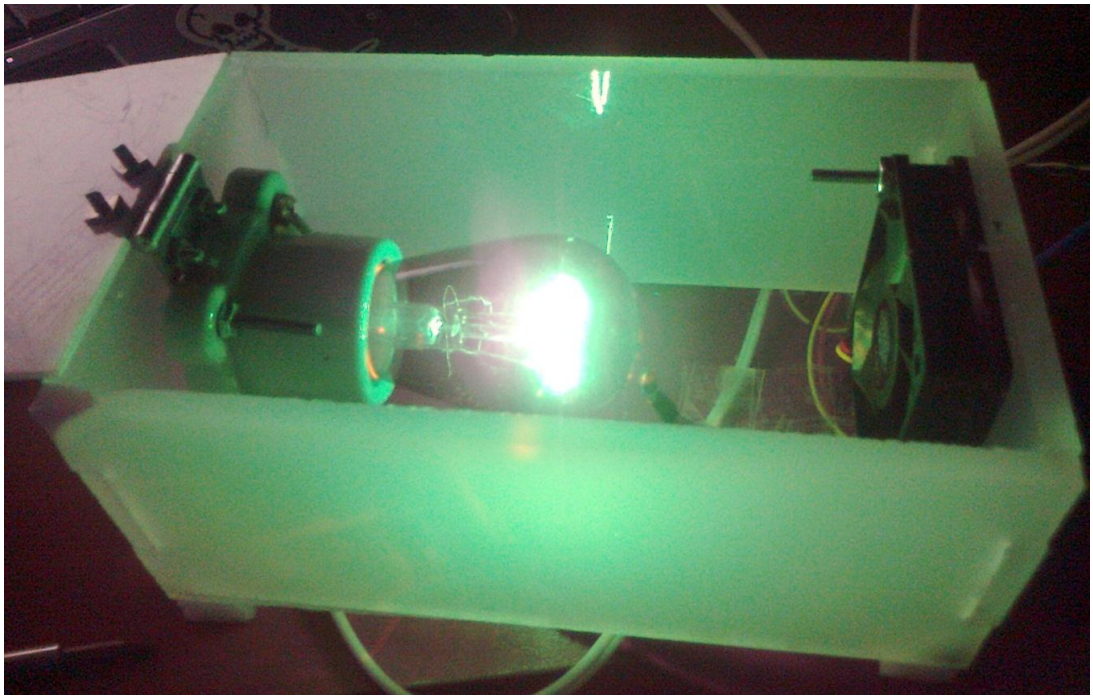
- This renders the serial port available for other sessions or the IDE environment

Una vez entendido lo anterior y puesto a prueba, nos proponemos a realizar la programación en matlab, para poder hacer la lectura del sensor LM35.

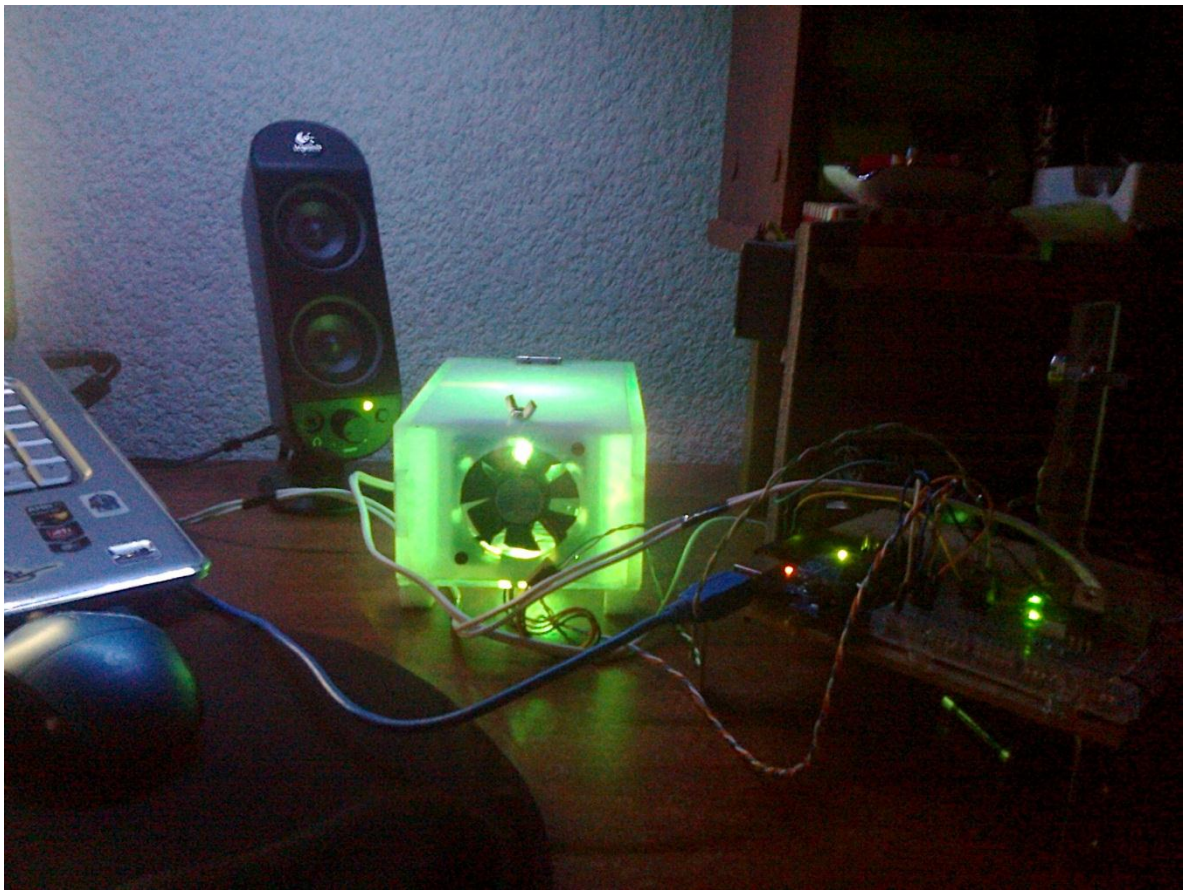
```
Programa
%Graficar datos
clc
disp('Captura de Datos mediante la lectura del puerto de ADC del Arduino
UNO')
disp('Esperando conexión....')
%Conectamos el Arduino
a=arduino('COM5');
disp('Arduino listo!!!')
%encendemos el ventilador
a.pinMode(9,'output')%Ventilador
a.pinMode(7,'output')%Foco
a.digitalWrite(9,1)
a.digitalWrite(7,0)
%Creamos la variable de muestras a tomar en tiempo real
Muestras=input('Dame el numero de muestras a capturar = ');
pause on %Habilitamos pause
%disp('Esperamos 10s');
%pause(10)
a.digitalWrite(9,0)
a.digitalWrite(7,1)
disp('Encendemos el foco');
%Creacion de la ventana de la Grafica
figure('Name','Lectura del sensor LM35')
title('Grafica de la temperatura Muestreada');
xlabel('Número de muestra por segundo');
ylabel('Voltaje (V)');
ylim([0 5.1]);
xlim([0 Muestras]);
grid on
hold on
%declaramos nuestra variable donde se guardaran los datos par despues
%graficarlos
Captura=zeros(1,Muestras+1);
%Inicia un bucle finito para la captura de datos
for i=0:Muestras
val=a.analogRead(0);
y=(val(1))*5/1024;
plot(i,y,'r*'),figure(gcf)
Captura(i+1)=y;
drawnow
pause (1)
end
a.digitalWrite(7,0)
hold off
pause(1)
%Graficamos los datos guardados
plot(Captura)
title('Grafica de la temperatura Muestreada');
xlabel('Número de muestra');
ylabel('Voltaje (V)');
grid on
delete(a);
disp('Termino la Captura de Datos')
```

Una vez realizado el código procedemos a realizar la estructura donde se realizaran las pruebas.

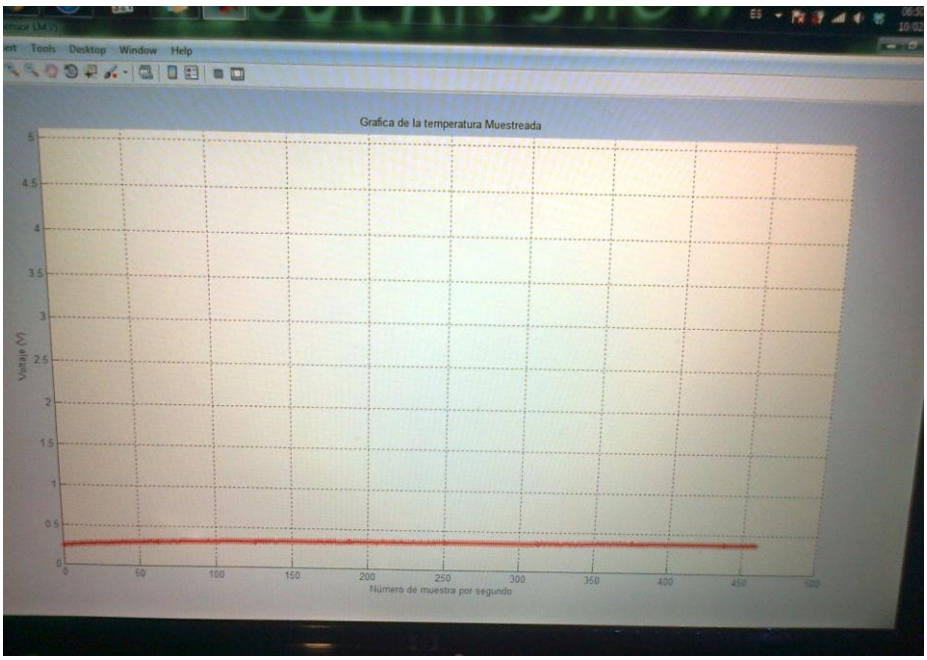
Vista de la estructura con el soquet donde se colocara el foco y el sistema de enfriado.



Vista del sistema Funcionando.

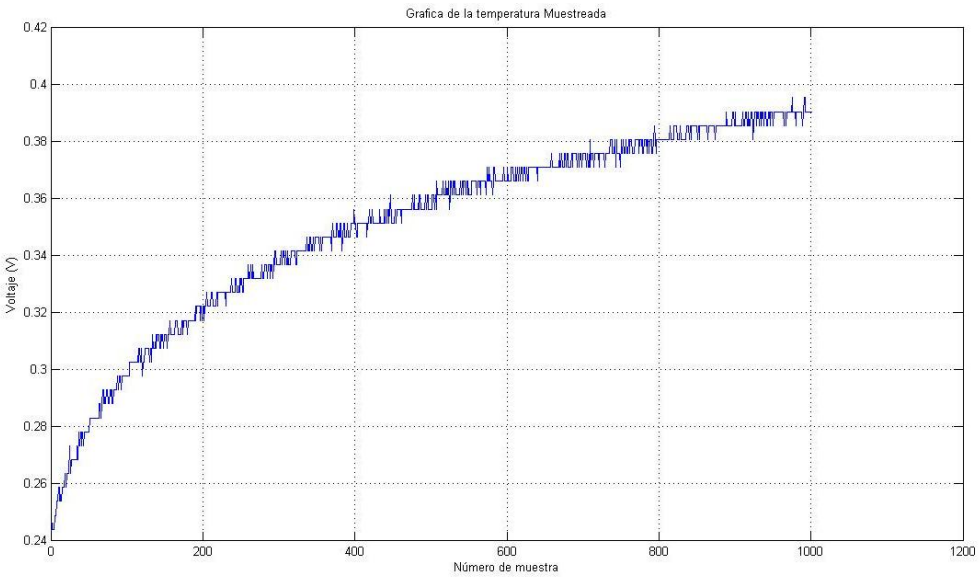


Otra Vista del sistema.



Vista de la captura de datos por segundo del sensor.

Resultados obtenidos



Grafica obtenida, aquí se observa que nuestra grafica se mantiene estable hasta los mil segundo.

Ahora apartir de los datos capturados obtenemos nuestra Funcion de transferencia. Nuestro sistema se comporta como un sistema de primer orden.

$$\frac{dc(t)}{dt} + a_0 c(t) = b_0 r(t)$$

Donde la Funcion de Transferencia es:

$$\frac{C(s)}{R(s)} = \frac{b_0}{s + a_0}$$

Reordenando los términos también se puede escribir como:

$$\frac{C(s)}{R(s)} = \frac{K}{\tau s + 1}$$

Donde K, es la ganancia en estado estable:

$$K = \frac{b_0}{a_0}$$

τ es la constante de tiempo del sistema.

$$\tau = \frac{1}{a_0}$$

Donde el valor s, se denomina polo:

$$s = -a_0 = -\frac{1}{\tau}$$

Con estos conceptos podemos ahora realizar la función de transferencia del sensor. El desarrollo a seguir es el siguiente:

1.- Se define la ganancia en estado estable

$$K = \frac{Temperatura\ en\ estado\ estable}{Voltaje\ de\ Entrada} = \frac{39° - 24°}{5\ V} = 3$$

2.- Se define la constante de tiempo. Usando el criterio del 2% de error, se determina el tiempo que tarda la salida en alcanzar un 98% de su valor, se divide entre 4 y se obtiene la constante de tiempo.

$$\tau = \frac{1000seg}{4} = 250$$

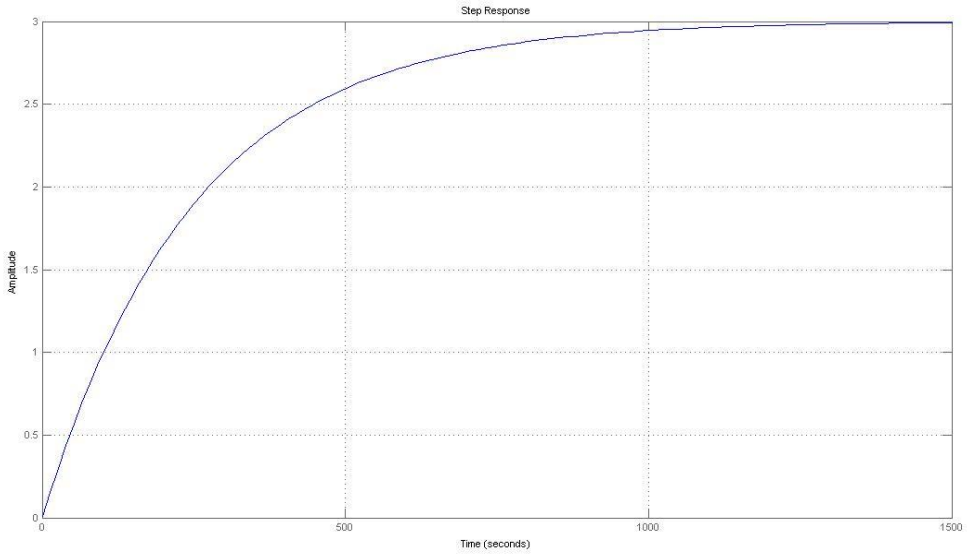
3.- Por ultimo se sustituye en la forma:

$$G(s) = \frac{K}{\tau s + 1}$$

4.- La función de transferencia que relaciona la temperatura con el voltaje es

$$G(s) = \frac{3}{250s + 1}$$

5.- La grafica obtenida, aplicando una entrada escalón unitario es el siguiente.



Vista de la Grafica obtenida de la función de transferencia.

Conclusiones

Nuevamente nos vuelve a sorprender matlab y la tarjeta Arduino, mediante esta práctica tuvimos la oportunidad de poder usar la tarjeta Arduino UNO como una tarjeta de adquisición de datos y a la vez con la programación adecuada pudimos graficar los valores leídos o capturados por el puerto analógico del Arduino y obtener los valores necesarios para poder caracterizar el sistema sensor.