

**PROFESOR:**

**INTEGRANTE:**

**Espinosa Sánchez Daniel Antonio.**

**ING. EN SISTEMAS COMPUTACIONALES**

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
Ingeniería en Sistemas Computacionales		SCC-1023	Sistemas Programables

PRÁCTICA No.	LABORATORIO	SALÓN DE CLASE	DURACIÓN (HORA)
3	Sensor de Temperatura	4701	2:00 HRS

1	INTRODUCCIÓN
<p>Continuando con el desarrollo de la unidad uno, se nos ha pedido programar el funcionamiento de un sensor de temperatura utilizando el software Arduino y Proteus, este tendrá que mostrar en la pantalla LCD la temperatura que ha sido asignada (medida) por el sensor.</p>	

2	OBJETIVO (COMPETENCIA)
<ul style="list-style-type: none"> <li>Realiza una práctica en Proteus utilizando un sensor de temperatura que nos arroje la medición en una pantalla LCD 16x2.</li> </ul>	

3	CÓDIGO
---	--------

```

#include<OneWire.h>
#include<DallasTemperature.h>
#include<LiquidCrystal.h>
#define ONE_WIRE_BUS 8
/*inicializar las librerias*/
OneWire oneWire (ONE_WIRE_BUS);
DallasTemperature sensor (&oneWire);
const int rs=12, en=11, d4=5, d5=4, d6=3, d7=2;
/*declarar las variables*/
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void setup() {
  Serial.begin(9600);
  sensor.begin();
  lcd.begin(16,2);
}
void loop() {
  sensor.requestTemperatures(); // requerimos la
  temperatura medida por el sensor
  lcd.clear(); // limpiamos pantalla lcd
  lcd.setCursor(0,0); // indicacion de la
  coordenada donde vamos a escribir
  lcd.print("Temperatura (C)"); // escribimos
  temperatura (C) en la pantalla lcd
  lcd.setCursor(7,1);
  lcd.print(sensor.getTempCByIndex(0)); //
  escribimos el valor obtenido en el sensor
  delay (500);
}

```

4	PROCEDIMIENTO (DESCRIPCIÓN)	
<p style="text-align: center;"><b>EQUIPO NECESARIO</b></p> <ul style="list-style-type: none"> <li>• Computadora o laptop</li> <li>• SO: Windows 8 o superior</li> <li>• Software Proteus 8</li> <li>• Software Arduino 1.8</li> </ul>	<p style="text-align: center;"><b>MATERIAL DE APOYO</b></p> <ul style="list-style-type: none"> <li>• Video tutorial</li> </ul>	

**B****DESARROLLO DE LA PRÁCTICA****1. Realizar la configuración del código**

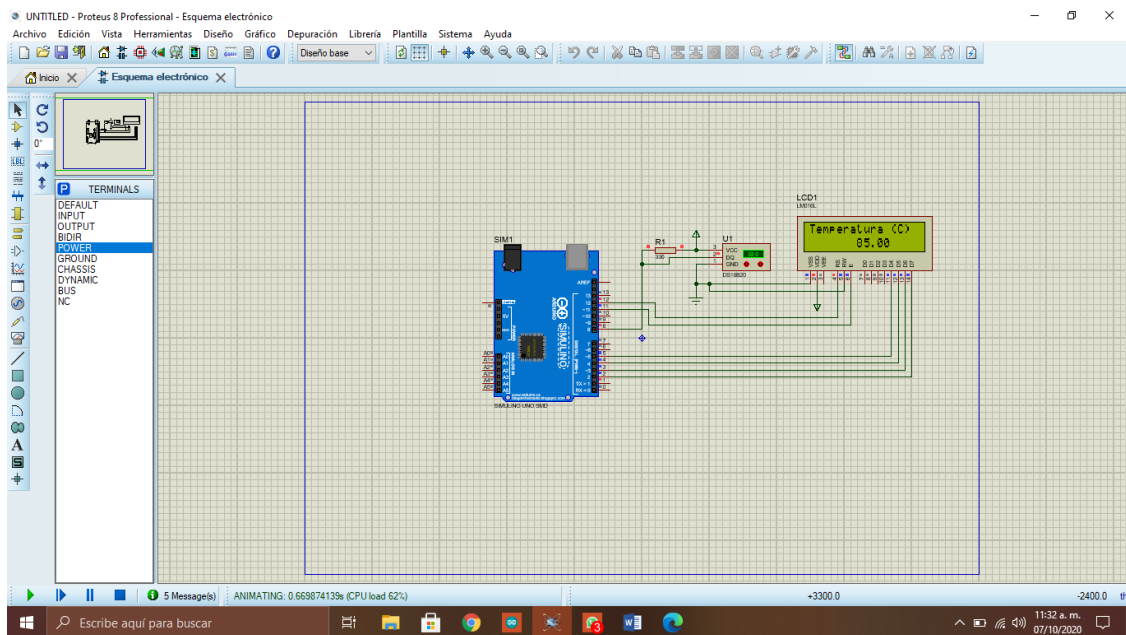
- Tendremos que abrir el programa “Arduino”, posteriormente procederemos a implementar los complementos (librerías) necesarios para el correcto funcionamiento del sensor.
- Una vez agregados dichos complementos procederemos a llamarlos y a su vez agregaremos la declaración de las variables.
- Por ultimo realizamos la configuración y programado sobre las acciones que realizara dichos complementos, en este caso: sensor de temperatura y LCD.

**2. Realizar el modelado en proteus**

- Primero ejecutaremos el software; posteriormente le daremos “click” en el apartado de “esquemas”, ahí procederemos agregar los siguientes competes: una resistencia (330 ohmz), un simulino SMD, una tierra, 2 de energía, un LCD y sensor de temperatura.
- A continuación, procederemos a colocar cada uno de los componentes con una cierta distancia entre cada uno, esto con el fin de simplificar los enlaces (conexiones).
- Por último, daremos doble click en simulino SDM, buscaremos el apartado “program file” y ahí pegaremos la dirección. HEX que copiamos del software arduino.

**3. Ejecución del programa**

- Por ultimo nos posicionaremos en la parte inferior derecha del programa, ahí nos encontraremos con un icono de “Play” lo presionamos; con eso ya estaríamos ejecutando el programa donde podremos observar su funcionamiento

**C CÁLCULOS Y REPORTE**

Respecto a esta práctica, no ha sido necesario realizar algún agregado o modificación al mismo, dado que la misma está bien desarrollada y complementada.

## **5 RESULTADOS Y CONCLUSIONES**

Como se puede observar en la captura de pantalla, dicho programa cumple con los parámetros establecidos al inicio de la práctica, demostrando su correcto funcionamiento a través del simulador "Proteus". A si mismo se puede observar los componentes necesarios que dicho software solicita para poder ejecutar las acciones programadas.

Conclusión:

Dicha práctica nos ayuda a familiarizarnos con el software Arduino, aprendiendo a configurar cada uno de los apartados y permitiéndonos desarrollar nuevas acciones para una misma problemática.

Por ultimo cabe agregar que cada vez es más entendible el lenguaje y configuración de los diferentes parámetros y complementos.