Grupo 3

PROYECTO TERMOMETRO DIGITAL

PRESENTADO POR:

JENNY CAROLINA CASAS V

JOVANY VARGAS GONZALEZ

JUAN CAMILO RINCON

PRESENTADO A:

Liliana Rodriguez Amorocho

ASIGNATURA: MICROCONTROLADORES Y MICROPROCESADORES

PROGRAMA INGENIERIA DE TELECOMUNICACIONES

UNIVERSIDAD COOPERATIVA DE COLOMBIA

BOGOTÁ

AÑO – 2013

OBJETIVOS

GENERAL

✓ **Realizar la implementación de un termómetro digital, aplicando los conocimientos adquiridos en el desarrollo de la asignatura.**

ESPECÍFICOS

✓ **Desarrollar diagrama de flujo del paso a paso del funcionamiento del nuestro proyecto en este caso termómetro.**

✓ **Realizar código assembler o código en C del termómetro digital.**

✓ **Simular en proteus el funcionamiento del termómetro digital.**

✓ **Comprender el funcionamiento del termómetro digital.**

INTRODUCCION

En el presente informe se explica el diseño y desarrollo para implementar un termómetro digital utilizando el micro controlador PIC16F877A, fabricado por Microchip, el cual se programará en el lenguaje ensamblador del micro controlador para el funcionamiento del proyecto.

La importancia de realizar un termómetro digital es debido a que es muy fácil realizar medidas de la temperatura con un sistema de adquisición de datos, pero la realización de medidas de temperatura exactas y repetibles no es tan fácil.

La temperatura es un factor de medida engañoso debido a su simplicidad. A menudo pensamos en ella como un simple número, pero en realidad es una estructura estadística cuya exactitud y repetitividad pueden verse afectadas por la masa térmica, el tiempo de medida, el ruido eléctrico y los algoritmos de medida.

La temperatura es difícil de medir con exactitud aún en circunstancias óptimas, y en las condiciones de prueba en entornos reales es aún más difícil. Entendiendo las ventajas y los inconvenientes de los diversos enfoques que existen para medir la temperatura, resultará más fácil evitar los problemas y obtener mejores resultados.

MATERIALES TERMÓMETRO DIGITAL

MATERIALES A UTILIZAR	CANTIDAD

COMPUTADOR	1
MPLAB IDE v88 o superior.	1
PROTEUS v7.7 o superior.	1
PICKIT2	1
QUEMADOR	1
SENSOR DE TEMPERATURA LM35	1
PIC 16F877A	1
CRISTAL 4MHZ	1
LCD 16X2	1
RESISTENCIA	1

TABLA 1.

FUNDAMENTACIÓN TEÓRICA

TERMÓMETRO DIGITAL

El termómetro digital es aquel que, valiéndose de dispositivos transductores, utilizan luego circuitos electrónicos para convertir en números las pequeñas variaciones de tensión obtenidas, mostrando finalmente la temperatura en un visualizador. Los termómetros digitales suelen medir la temperatura de manera más rápida y precisa. Vienen en muchos tamaños y formas, y están disponibles en la mayoría de los supermercados y farmacias, a varios precios. Siempre se deben de leer las instrucciones del fabricante a fin de determinar los métodos para los que está diseñado el termómetro.

SENSOR LM35

El LM35 es un sensor de temperatura con una precisión calibrada de 1ºC y un rango que abarca desde -55º a +150ºC. 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 - Vout - GND.

PIC 16F877A

El PIC16F877A es un microcontrolador con memoria de programa tipo FLASH, lo que representa gran facilidad en el desarrollo de prototipos y en su aprendizaje ya que no se requiere borrarlo con luz ultravioleta como las versiones EPROM, sino que permite reprogramarlo nuevamente sin ser borrado con anterioridad.

El PIC16F877A es un micro controlador de Microchip Technology fabricado en tecnología CMOS, su consumo de potencia es muy bajo y además es completamente estático, esto quiere decir que el reloj puede detenerse y los datos de la memoria no se pierden.

PROCEDIMIENTO

Realizamos el montaje del circuito en la protoboard, usando dos componentes principales los cuales fueron el sensor de temperatura LM35 y el PIC 16F877A.

Usamos un LCD de 16X2, para mostrar la temperatura en pantalla, además realizamos el diagrama de flujo, código assembler y la simulación en proteus por lo cual se puedo evidenciar el funcionamiento del termómetro digital antes de realizar el montaje.

En forma breve, la función del termómetro puede explicarse a través de los siguientes pasos:

1. Con un programa llamado "MPLAB", se establecen una serie de instrucciones a la PIC (16f877A) por medio de códigos especiales y con un programador que sirve de interfaz entre la computadora y el PIC y de acuerdo a las especificaciones se pueden mandar las instrucciones.

2. La PIC convierte los datos de temperatura en información de tipo eléctrica, para que pueda ser procesada por el display y de este modo se lleve a cabo la lectura de la temperatura.

DISEÑO DEL TERMÓMETRO DIGITAL

El termómetro digital será desarrollado de acuerdo al siguiente diagrama a bloques:

Gráfico 1. Diagrama de bloques

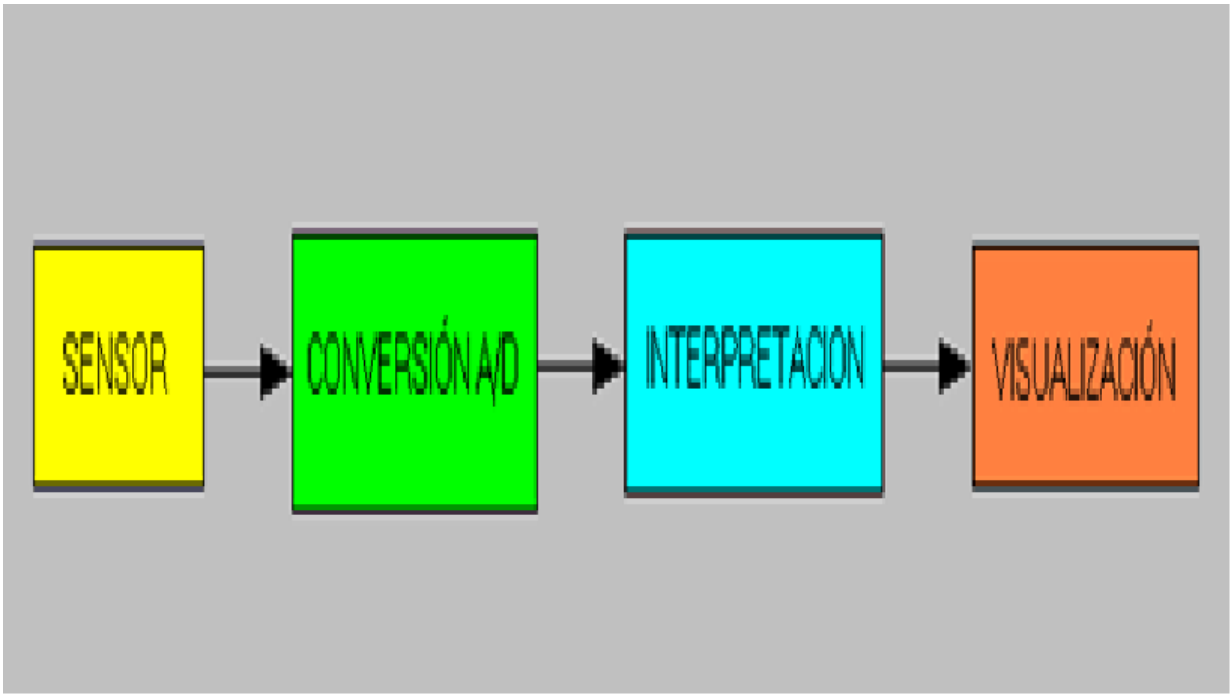


DIAGRAMA DE FLUJO

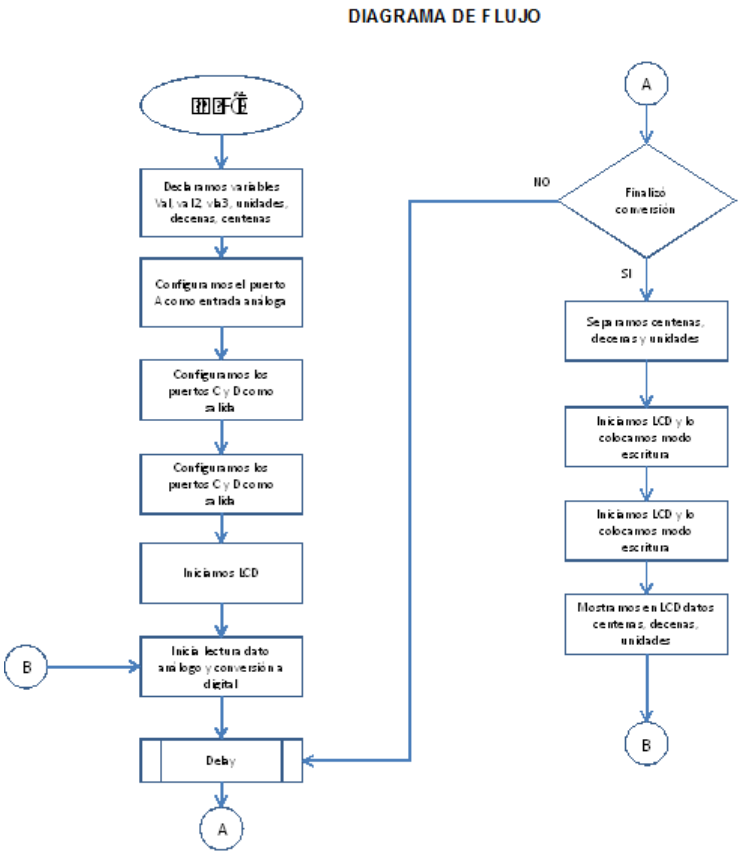


Gráfico 2.

CÓDIGO DEL PROGRAMA

```
processor 16f877a

list p=16f877a

include<p16f877a.inc>

;Variable para el DELAY del ADC

val equ h'20'


;Variables para el DELAY del ENABLE LCD

val2 equ 0x30

val1 equ 0x31


;Variables para separar el resultado de ADC

; en valor BCD

Unidades equ 0x32

Decenas equ 0x33

Centenas equ 0x34

Resto equ 0x35
```

```
org 0h
goto INICIO
org 05h

INICIO:
clrf PORTA ;Limpia el puerto A
clrf PORTC ;Limpia el puerto C
clrf PORTD ;Limpia el puerto D
bsf STATUS,RP0
bcf STATUS,RP1 ;Cambio la banco 1
;Configuración de puertos C-D para el LCD
clrf TRISC ;Configura PORTC como salida
clrf TRISD ;Configura PORTD como salida
;Configuración del puerto A para ADC
movlw 00h
movwf ADCON1 ;Configura puerto A y E como analógicos
movlw 3fh
movwf TRISA ;Configura el puerto A como entrada
movlw h'0'
bcf STATUS,RP0 ;Regresa al banco 0
;Inicio del programa
START
call START_LCD ;Inicializa LCD
goto START_ADC ;Comienza la lectura del Conv. A/D
;Inicia LCD
START_LCD:
bcf PORTC,0 ; RS=0 MODO INSTRUCCION
movlw 0x01 ; El comando 0x01 limpia la pantalla en el LCD
movwf PORTD
call COMANDO ; Se da de alta el comando
movlw 0x0C ; Selecciona la primera línea
movwf PORTD
call COMANDO ; Se da de alta el comando
movlw 0x3C ; Se configura el cursor
movwf PORTD
call COMANDO ; Se da de alta el comando
bsf PORTC, 0 ; Rs=1 MODO DATO
return
;Rutina para enviar un dato
ENVIA:
bsf PORTC, 0 ; RS=1 MODO DATO
```

```
call COMANDO    ; Se da de alta el comando

return

;Rutina para enviar comandos

COMANDO:

bsf PORTC, 1      ; Pone la señal ENABLE en 1

call DELAY2      ; Tiempo de espera

call DELAY2

bcf PORTC, 1     ; ENABLE=0

call DELAY2

return

;Rutina para limpar pantalla LCD

ERASE_LCD

bcf PORTC,0      ; RS=0 MODO INSTRUCCION

movlw 0x01       ; El comando 0x01 limpia la pantalla en el LCD

movwf PORTD

call COMANDO     ; Se da de alta el comando

bsf PORTC, 0     ; Rs=1 MODO DATO

return

;Configuración Convertidor A/D

START_ADC

movlw b'11000001' ;Configuración ADCON0

movwf ADCON0     ;ADCS1=1 ADCS0=1 CHS2=0 CHS1=0 CHS0= GO/DONE=0 - ADON=1


CICLO: bsf ADCON0,2      ;Conversión en progreso GO=1

call DELAY1       ;Espera que termine la conversión

ESPERA btfsc ADCON0,2    ;Pregunta por DONE=0? (Terminó conversión)

goto ESPERA       ;No, vuelve a preguntar

movf ADRESH,0     ;Si

;Rutina que muestra temperatura

PRINT_TEMP

call ERASE_LCD    ;Limpia LCD

movlw 'T'

movwf PORTD

call ENVIA

movlw 'E'

movwf PORTD

call ENVIA

movlw 'M'

movwf PORTD

call ENVIA

movlw 'P'
```

```
movwf PORTD
call ENVIA
movlw '='

movwf PORTD
call ENVIA
movlw ' '
movwf PORTD
call ENVIA

call READ_TEMP    ;Llamada a rutina que obtine el
                  ;valor de la temperatura a partir
                  ;del  resultado del Conv a/D

movf Centenas,W   ;Imprime el dígito de las centenas
movwf PORTD
call ENVIA
movf Decenas,W    ;Imprime el dígito de las decenas
movwf PORTD
call ENVIA
movf Unidades,W   ;Imprime el dígito de las unidades
movwf PORTD
call ENVIA
movlw ' '
movwf PORTD
call ENVIA
movlw h'DF'       ;Imprime el simbolo "°"
movwf PORTD
call ENVIA
movlw 'C'
movwf PORTD
call ENVIA

goto CICLO        ;Repite el ciclo de lectura ADC

;Rutina que obtine el valor de la temperatura
;a partir del  resultado del Conv a/D
READ_TEMP:
clrf Centenas
clrf Decenas
clrf Unidades
```

```
movf ADRESH,W
addwf ADRESH,W     ;Dupilca el valor de ADRESH para
;obtener un valor de temperatura real aprox
movwf Resto        ;Guarda el valor de ADRESH en Resto
;Comienza el proceso de otención de valores BCD
;para Centenas, Decenas y unidades atraves de restas
;sucesivas.
CENTENAS1
movlw d'100'      ;W=d'100'
subwf Resto,W     ;Resto - d'100' (W)
btfss STATUS,C    ;Resto menor que d'100'?
goto DECENAS1     ;SI
movwf Resto       ;NO, Salva el resto
incf Centenas,1   ;Incrementa el contador de centenas BCD
goto CENTENAS1    ;Realiza otra resta
DECENAS1
movlw d'10'       ;W=d'10'
subwf Resto,W     ;Resto - d'10' (W)
btfss STATUS,C    ;Resto menor que d'10'?
goto UNIDADES1    ;Si
movwf Resto       ;No, Salva el resto
incf Decenas,1    ;Incrementa el contador de centenas BCD
goto DECENAS1     ;Realiza otra resta
UNIDADES1
movf Resto,W      ;El resto son la Unidades BCD
movwf Unidades
;clrf Resto
;Rutina que obtiene el equivalente en ASCII
OBTEN_ASCII
movlw h'30'
iorwf Unidades,f
iorwf Decenas,f
iorwf Centenas,f
return


;Rutina que genera un Delay de 20 microSeg aprox.
;para el Conv. A/D
DELAY1:
movlw h'30'
```

```
movwf val
Loop   decfsz val,1
goto Loop
return

;Subrutina de retardo para ENABLE_LCD
DELAY2:
movlw 0xFF
movwf val1
Loop1:
movlw 0xFF
movwf val2
Loop2:
decfsz val2,1
goto Loop2
decfsz val1,1
goto Loop1
return
end
```

ESQUEMA DE CONEXIÓN

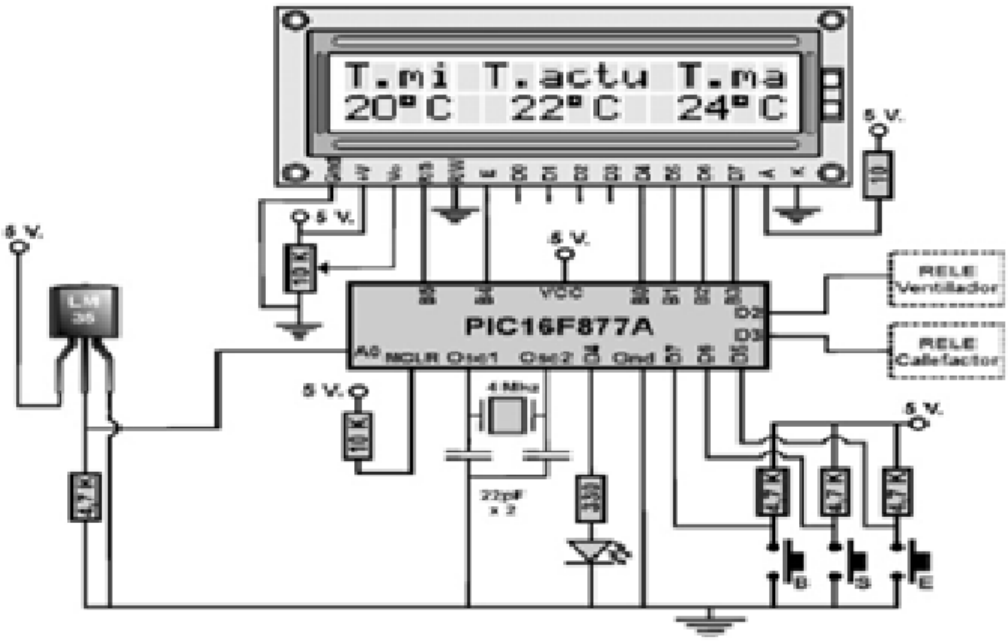


Gráfico 3.

SIMULACIÓN TEMPERTURA

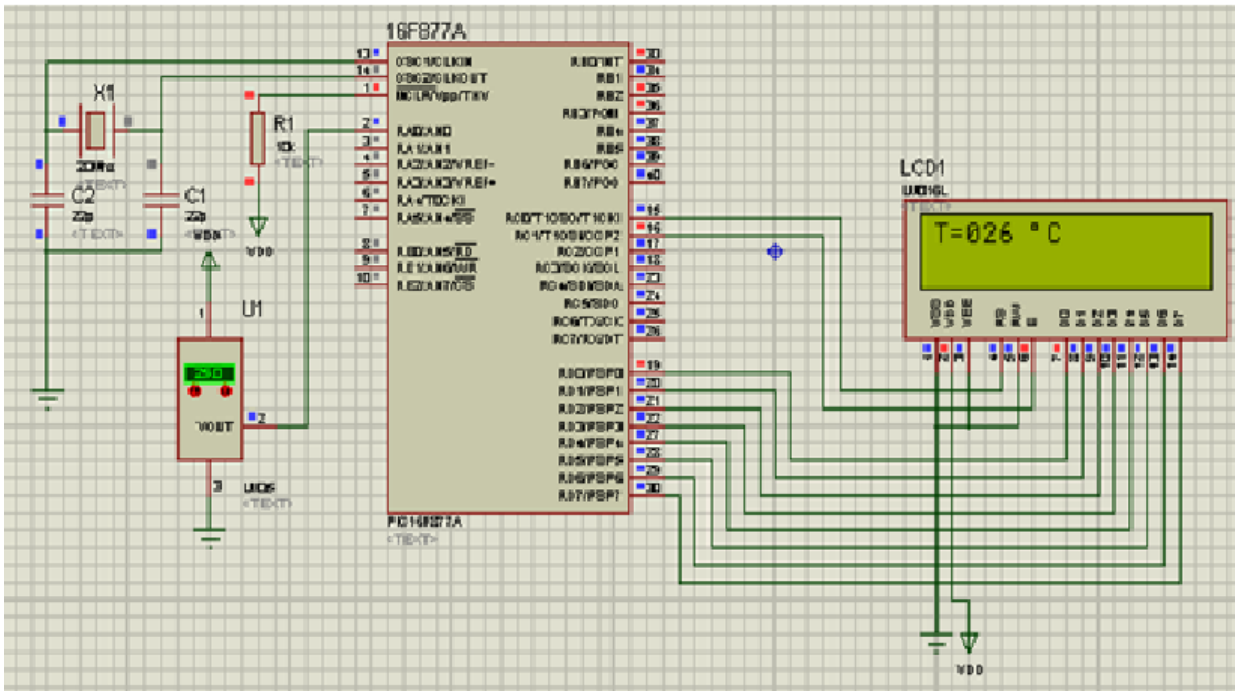


Gráfico 4. Montaje en Proteus

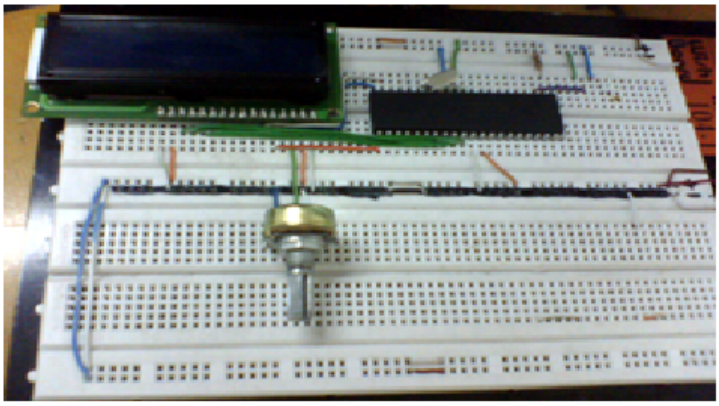


Gráfico 5.

DATASHEET PIC 16F877A

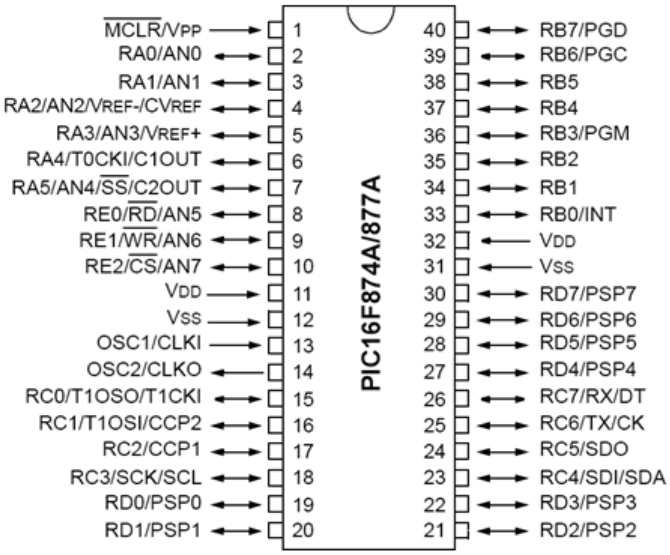


Gráfico 6.

DATASHEET LM35

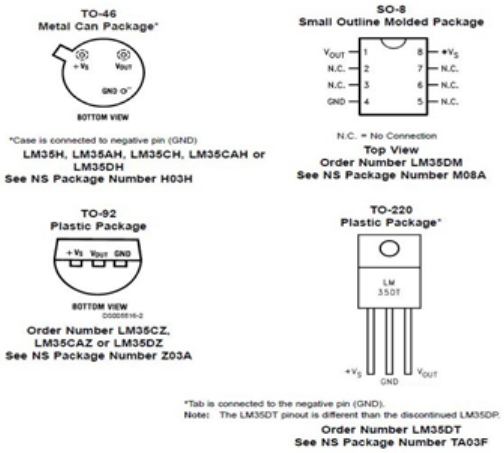


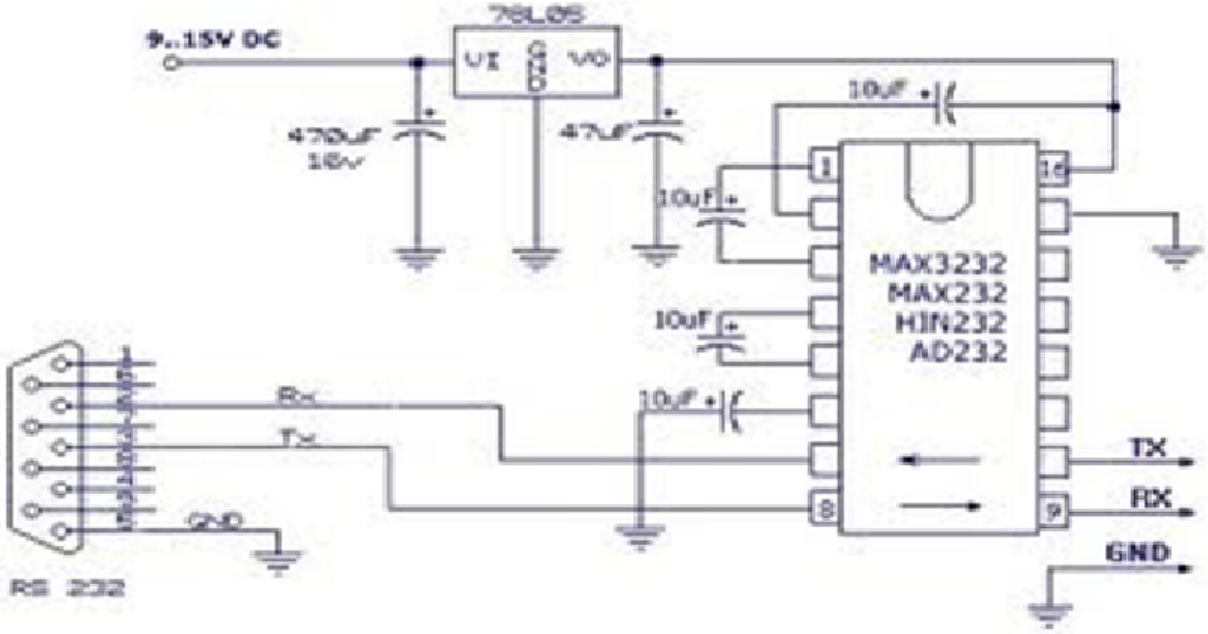
Gráfico 7.

FUNCIONAMIENTO

Para esta práctica necesitamos un sensor de temperatura LM35, este dispositivo presenta en su pin OUT una variación de 10 mV, por grado centígrado, su alimentación puede ser de 4 a 30 Voltios, y su rango de temperatura a medir entre -55°C hasta 150 °C.

Para medir la temperatura se conecta el pin out del LM35 al puerto A.0, el cual está configurado como conversor A/D a 10 bits, este valor se almacena en la variable dato que tiene capacidad de 2 bytes (16 bits), el cual se lo divide para 128, debido a que la variable del conversor A/D de 10 bits lo presenta en 16 bits, de la siguiente manera:1111111111000000, los 6 bits que contienen ceros se los debe ignorar, ya que sólo necesitamos los 8 bits del 1er byte más 2 bits del segundo byte, si este dato lo dividimos para 64 conseguiremos eliminar los 6 bits que corresponde a los ceros, de esta manera tendremos el dato a 10 bits, es decir el C A/D mostraría como valor máximo 1024, esto es una resolución de 5 mV, pero como el LM35 tiene incrementos de 10 mV, debemos bajar la resolución a 9 bits y eso se consigue dividiendo para 128, lo cual elimina 7 bits del 2do byte, de esta manera el valor más alto sería 512, esto es lo más cercano a la escala del LM35. El proyecto funciona de la siguiente manera: si la temperatura permanece entre20°C y 24°C ninguno de los relés se activa, pero si la temperatura no se encuentra entre estos 2 rangos, se activa el relé que le corresponde, sea para calentar o enfriar el ambiente, si deseamos modificar los rangos de temperatura, presionamos el pulsador E, con los otros2 botones aumentamos o disminuimos la temperatura mínima a comparar, y una vez que estemos de acuerdo presionamos la tecla E nuevamente, luego nos pide programar la temperatura máxima, procedemos igual que el caso anterior y cuando presionemos la tecla E, parpadeará tres veces el led, indicando que los nuevos valores ya fueron guardados en la memoria no volátil.

Conexión con puerto Serial para el Hiperterminal

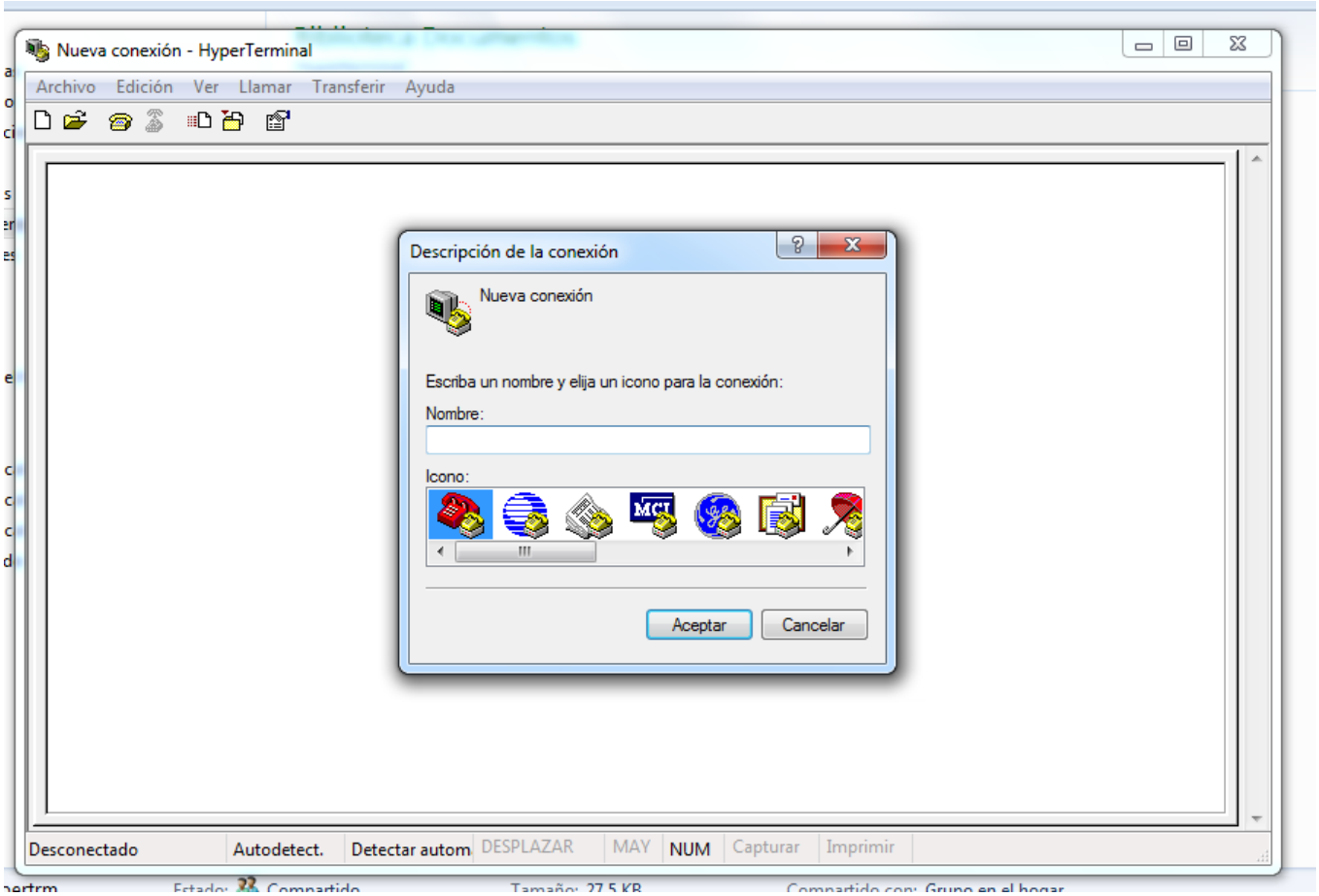


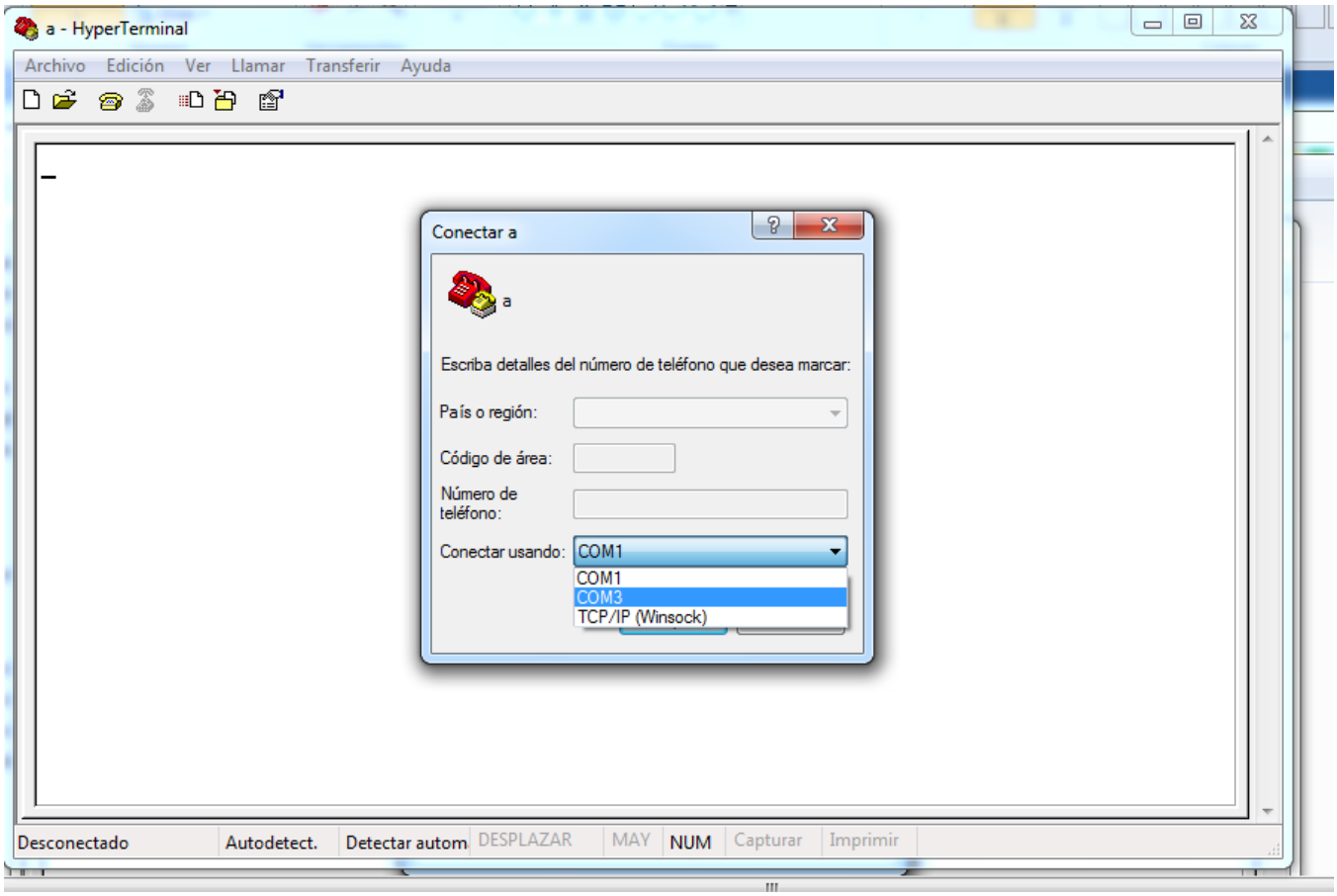
Programa en C

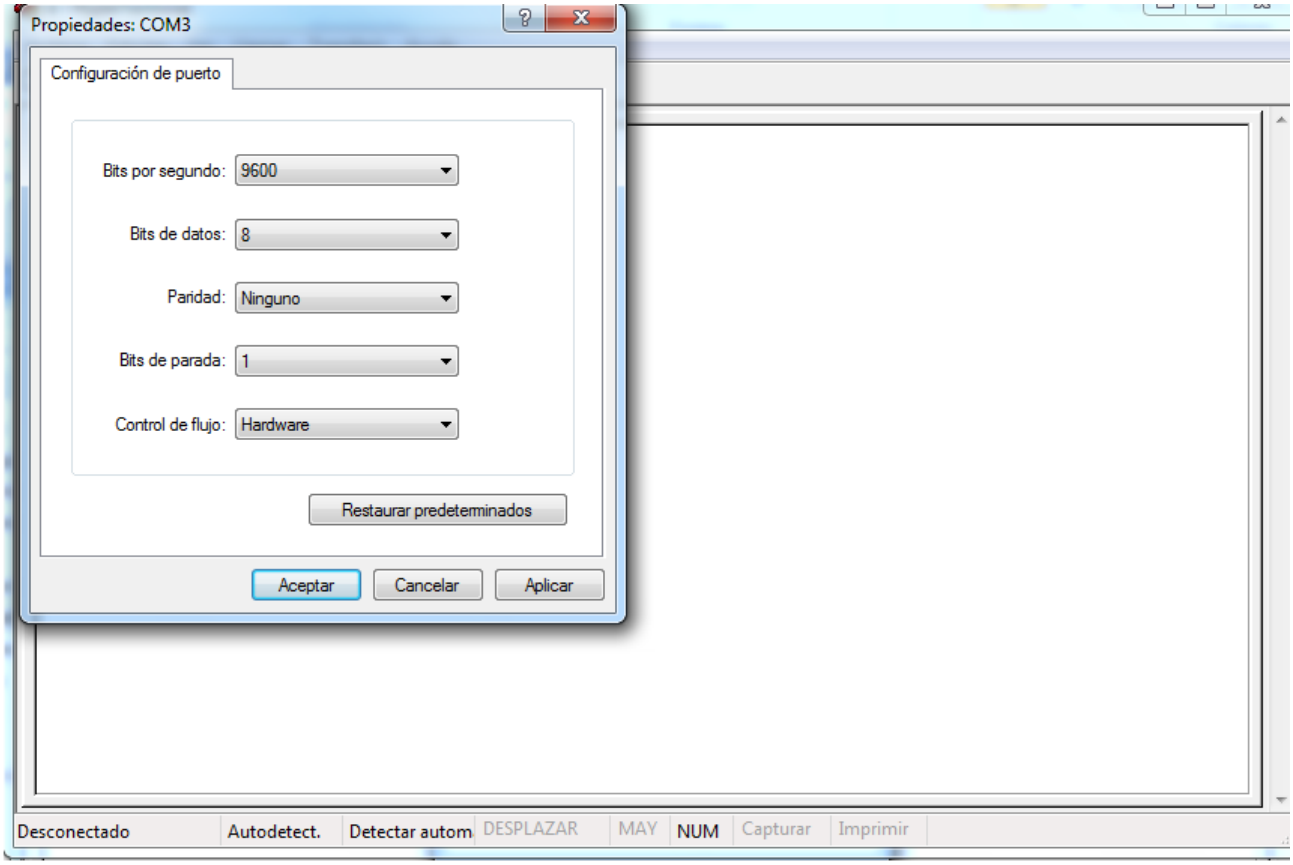
```
#include "16f877a.h"
#device adc=10
#FUSES xt,nowdt
#use delay(clock=4Mhz)
#use RS232(BAUD=9600,BITS=8,PARITY=N,XMIT=PIN_C6,RCV=PIN_C7)
#include "LCD.C"

void main(){
int16 q;
float p;
setup_adc_ports(AN0);
setup_adc(ADC_CLOCK_INTERNAL);
lcd_init();

while(1){
set_adc_channel(0);
```

```
delay_us(10);
q=read_adc();
p=5.0*q/1024.0;
printf("Voltaje:%0.1f\f ",p);
lcd_gotoxy(1,1);
printf(lcd_putc,"Voltaje:%0.1f\r ",p);
delay_ms(100);

if(p>3){
output_high(PIN_b0);
lcd_gotoxy(1,2);
printf(lcd_putc,"Alerta");
printf("Alerta");
}
else{
output_low(PIN_B0);
lcd_gotoxy(1,2);
printf(lcd_putc,"Normal");
}
}
}
```

CONECTANDO AL HIPERTERMINAL


Nueva conexión - HyperTerminal
Archivo Edición Ver Llamar Transferir Ayuda
Descripción de la conexión
Nueva conexión
Escriba un nombre y elija un icono para la conexión:
Nombre:
Icono:
Aceptar
Cancelar
Desconectado
Autodetect.
Detectar autom
DESPLAZAR
MAY
NUM
Capturar
Imprimir


a - HyperTerminal
Archivo Edición Ver Llamar Transferir Ayuda
Conectar a
a
Escriba detalles del número de teléfono que desea marcar:
País o región:
Código de área:
Número de teléfono:
Conectar usando:
COM1
COM1
COM3
TCP/IP (Winsock)
Desconectado
Autodetect.
Detectar autom
DESPLAZAR
MAY
NUM
Capturar
Imprimir



SIMULACION EN PROTEUS

Vídeo de YouTube

PROYECTO

Vídeo de YouTube

DIFICULTADES

Una de las mayores dificultades fue decidirnos por que proyecto podríamos seguir ya que en un principio pensamos en una alarma, pero era demasiado sencilla y no cumplía con las expectativas de la materia.

Trabajar en equipo cuesta un poco de trabajo cuando todos los integrantes tienen bastantes vacíos frente a un programa

Despues de solucionar estos inconvenientes el otro factor que incurrio en demoras fue la conexion serial al computador y el uso del hiperterminal

Aunque se quemaron 3 pic 16f877a se logro encontrar el corto en la protoboard y el fallo en el programa ya que se estaba configurando de manera análoga teniendo que estar en ese momento de manera digital

Fueron muchas las dificultades pero se logro completar las esectativas

CONCLUSIONES

✓ **Logramos concluir que los termómetros tienen cierto grado de error, por eso es importante conocer cuál es este grado y basarnos en promedios de los mismos para una mejor obtención de la lectura de temperatura.**

.

✓ **Por otra parte podemos decir que una desventaja del termómetro a base del sensor LM35 radica en que no es capaz de indicar una temperatura estable solamente, sino que por la sensibilidad del sensor este varía aunque sea muy poco el cambio de temperatura y puede haber complicaciones a la hora de la lectura sí llegase a ser mucha la diferencia del cambio, aunque esto ocurre muy pocas veces.**

✓ **Obtuvimos valores aproximados de la temperatura.**

✓ **Confirmamos la linealidad del sensor y su función como se observa claramente el incremento de 1ºC cada 10mV que cambia el sensor.**

WEBGRAFÍA

Articulo online disponible en:

http://ieupao.blogspot.com/2009/07/termometro-digital-con-lm35-y-pic.html.

Articulo online disponible en:

http://www.monografias.com/trabajos15/termometro-digital/termometro-digital.shtml.

Articulo online disponible en:

http://www.proyectoslibres.net/index.php?title=Term%C3%B3metro_con_16F877A

Manual de referencia LCD online disponible en: http://www.todorobot.com.ar/documentos/display.pdf.