



RA1000
Diseñar un experimento apropiadamente.
(Sensor de temperatura)

Alumno:
Andre Nicasio Romo

Carrera: Ingeniería Mecatrónica

Materia: Microcontroladores II

Profesor: María Teresa Orvañanos Guerrero

Fecha de entrega: 15 de abril del 2024
Universidad Panamericana de México, Ags.

Índice:

Planificacion:..... 3

Conexiones Electricas:.....4

Programacion:.....5

Implementacion:.....9

Planificación:

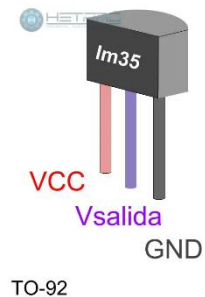
Antes de iniciar a diseñar un circuito se requiere conocer bien que es lo que se nos pide; primeramente, se nos está pidiendo que diseñemos un circuito el cual sea capaz de leer la temperatura ambiental, teniendo esto en cuenta lo primero que se me viene a la mente con la palabra “medición” son las lecturas del micro vía ADC, ya que anteriormente en clase hemos trabajamos con mediciones ADC.

Tomando esto en consideración, que herramientas del micro nos pueden ayudar a cumplir con el objetivo.

Ahora, haciendo una investigación (y por sugerencia de la maestra) se nos pide usar el sensor LM35 para realizar el experimento, Por lo que se procedió a investigar bien como es que funciona este sensor, navegando en la red nos topamos con:

LM35:

“El sensor LM35 es un sensor de temperatura muy popular por su fácil uso y variadas aplicaciones. No necesita de ningún circuito adicional para ser usado, se alimenta directamente con una fuente de 5V y entrega una salida de voltaje lineal analógica.”



SENSOR DE TEMPERATURA LM35 METÁLICO, 3 HILOS. (s/f).

microside. Recuperado el 15 de abril de 2024, de

<https://store.microside.com/productos/sensor-de-temperatura-lm35-metalico-3-hilos/>

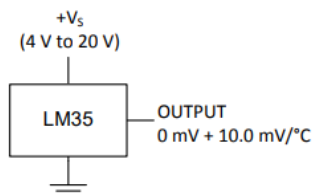
Lo que se puede entender la descripción anterior es que el LM35 consta de 3 partes (o pines) los cuales son un pin de voltaje de entrada, un pin de la salida de voltaje analógico y el pin de tierra, por lo que es fácil para nosotros intuir que del pin donde tomaremos la medida del ADC seria del pin de salida analógica.

Adentrándonos en la datasheet del sensor encontramos que el sensor LM35 soporta de 4V hasta 20V en el pin VCC (voltaje de entrada) y como máximo saca hasta 1500 mV en el pin de salida analógica, ya que la temperatura máxima que lee el sensor es de 150 grados centígrados, pero, ¿Cómo se sabe eso?

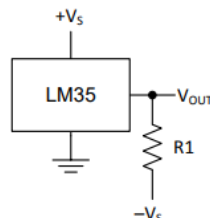
Relación lineal del sensor y Conexiones Electricas:

La señal de salida del sensor es directamente proporcional a la temperatura a la que este sujeto, ya que esta relación lineal es de $0+10\text{mV}/^\circ\text{C}$, donde dependiendo del tipo de conexión se puede obtener desde 2°C a 150°C y -55°C a 150°C , según la datasheet no requiere de ningún tipo de calibración adicional

**Basic Centigrade Temperature Sensor
(2°C to 150°C)**



Full-Range Centigrade Temperature Sensor



Choose $R_1 = -V_S / 50 \mu\text{A}$
 $V_{OUT} = 1500 \text{ mV at } 150^\circ\text{C}$
 $V_{OUT} = 250 \text{ mV at } 25^\circ\text{C}$
 $V_{OUT} = -550 \text{ mV at } -55^\circ\text{C}$

Features, 1., & Description, 3. (s/f). LM35 precision centigrade temperature sensors.

Www.ti.com. Recuperado el 15 de abril de 2024, de

<https://www.ti.com/lit/ds/symlink/lm35.pdf>

Planificación del código:

Conociendo las especificaciones del sensor y sus respectivas configuraciones se optó por elegir la configuración que más se adaptaría para nuestro experimento, en este caso sería la configuración de 2°C a 150°C , esto debido a que no hay forma de que encontremos un lugar que este a una temperatura de -55°C

Teniendo bien claro lo anterior, podemos deducir que la medición del sensor es meramente una medición de voltaje, algo así como lo que hemos venido haciendo en clase.

Aprovechando la función lineal de tenemos, podemos modificarla a nuestro favor para que la temperatura nos de con 1 punto decimal, gracias al hecho de que 10mV es un grado centígrado, por lo que si tomamos la lectura del ADC (Ya convertida a mV) y aplicandola con la función lineal tenemos:

$(ADC\text{mV})/(10\text{mV}/^\circ\text{C})$ nos da como resultado: $(ADC\text{ }^\circ\text{C})/10 = \text{Temperatura}$ (esto sería sin punto decimal)

Por lo que modificándola para tener un punto decimal seria simplemente multiplicando la lectura por 10, dándonos como resultado que la temperatura es igual a la lectura del ADC.

Programacion:

Como vimos anteriormente, el código que implementaremos será simplemente tomar la medición del ADC en mV y separar los dígitos de la lectura, ejemplo, para una temperatura de 25.3 grados la medición sería 253mV dando por hecho que el ultimo dígito será el dígito con valor del punto decimal.

Codigo:

```
/*
 * RA1000: Sensor Temperatura
 *
 * Created: 15/04/24
 * Author : Andre Nicasio Romo
 */

#define F_CPU 1000000UL
#include <avr/io.h> //Librería general del Microcontrolador
#include <stdint.h> //Para poder declarar variables especiales
#include <util/delay.h> //Para poder hacer delays
#include <avr/pgmspace.h>
#include "lcd.h"
#include <avr/interrupt.h> //Para poder manejar interrupciones

//PROTOTIPADO DE FUNCIONES PARA PODER UTILIZARLAS DESDE CUALQUIER "LUGAR"
//*****
uint8_t cero_en_bit(volatile uint8_t *LUGAR, uint8_t BIT);
uint8_t uno_en_bit(volatile uint8_t *LUGAR, uint8_t BIT);
//*****
unsigned long medicion_adc=0;
unsigned long medicion_mv=0;
uint8_t unidad, decima, centecima, milesima=0;

void Traba(volatile uint8_t *Lugar, uint8_t Bit){
    _delay_ms(50);
    while(cero_en_bit(&*Lugar, Bit)){
        _delay_ms(50);
    }
}

void digitos(int timeC, uint8_t *u, uint8_t *d, uint8_t *c, uint8_t *m){ //function para
//separar los dígitos
    *u= (timeC/1000); //valor de la unidad
    while(timeC>=1000){
        timeC-=1000;
    }
    *d= timeC/100;
    while(timeC>=100){
        timeC-=100;
    }
    *c= timeC/10;
    while(timeC>=10){
        timeC-=10;
    }
    *m=timeC;
}
```

```

void delete_seg(float seg){ //delay para segundos y fraccion.
    seg*=1000.00;
    uint8_t auxCont;

    int mseg= (int)seg;

    while(mseg>1000){
        auxCont=0;
        while (auxCont<4){

            _delay_ms(250);
            auxCont+=1;

        }

        mseg-=1000;
    }

    while(mseg>0){
        if(mseg>=250){
            _delay_ms(250);
            mseg-=250;
        }
        else if (mseg>=100){
            _delay_ms(100);
            mseg-=100;
        }
        else if (mseg>=10){
            _delay_ms(10);
            mseg-=10;
        }
        else{
            _delay_ms(1);
            mseg-=1;
        }
    }
}

void delay_s(uint8_t seg){ // delay para segundos enteros
    uint8_t auxCont;
    uint8_t aux_seg=0;

    while(aux_seg<seg){
        auxCont=0;

        while (auxCont<4){
            _delay_ms(250);
            auxCont+=1;
        }
        aux_seg+=1;
    }
}

```

```

void Conversion(){
    medicion_mv=((medicion_adc)*(5000))/1023;
    digitos(medicion_mv, &unidad, &decima, &centecima, &milesima);

    lcd_gotoxy(8,1);
    lcd_putc(unidad+48);
    lcd_putc(decima+48);
    lcd_putc(centecima+48);
    lcd_putc('.');
    lcd_putc(milesima+48);
    delete_seg(0.5); //Mini delay para que se visualice mejor
}

int main(void)
{
    sei();
    ADMUX=0b01000111; // Se seteo el adc con una precisión de 10bit, ya que la lectura
                      // debe ser precisa en cuestión de Mv, se leerá el puerto A07
    ADCSRA=0b10011011; // Sin interrupcion del adc
    lcd_init(LCD_DISP_ON);
    lcd_gotoxy(0,0);
    lcd_puts("Temperatura:");

    while (1)
    {
        ADCSRA|=(1<<6);
        while (uno_en_bit(&ADCSRA,6)){
            medicion_adc=ADC;
            Conversion();
        }
    }

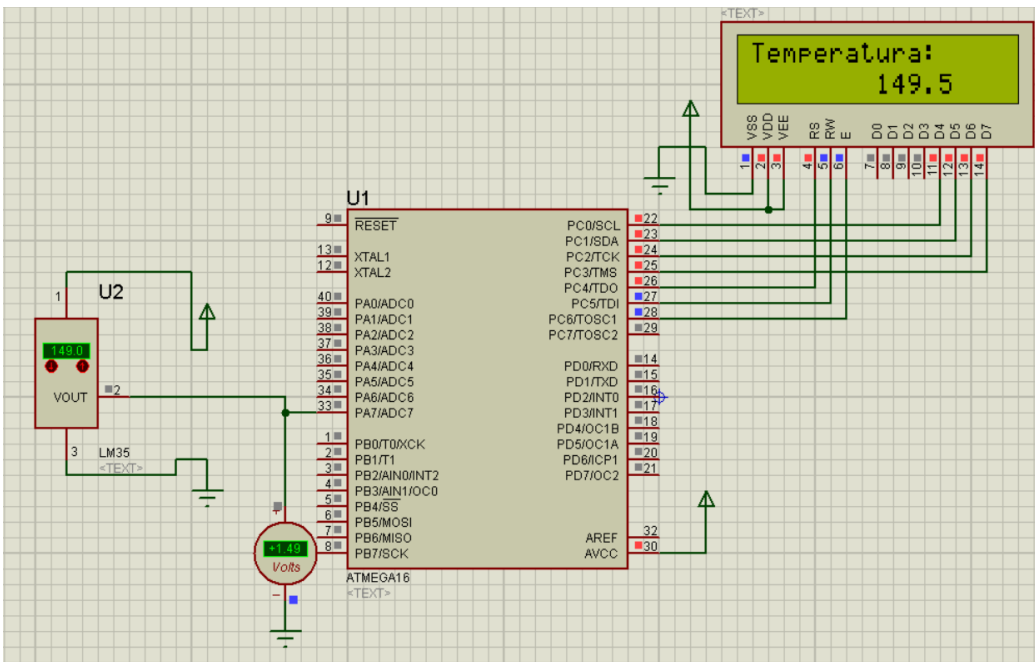
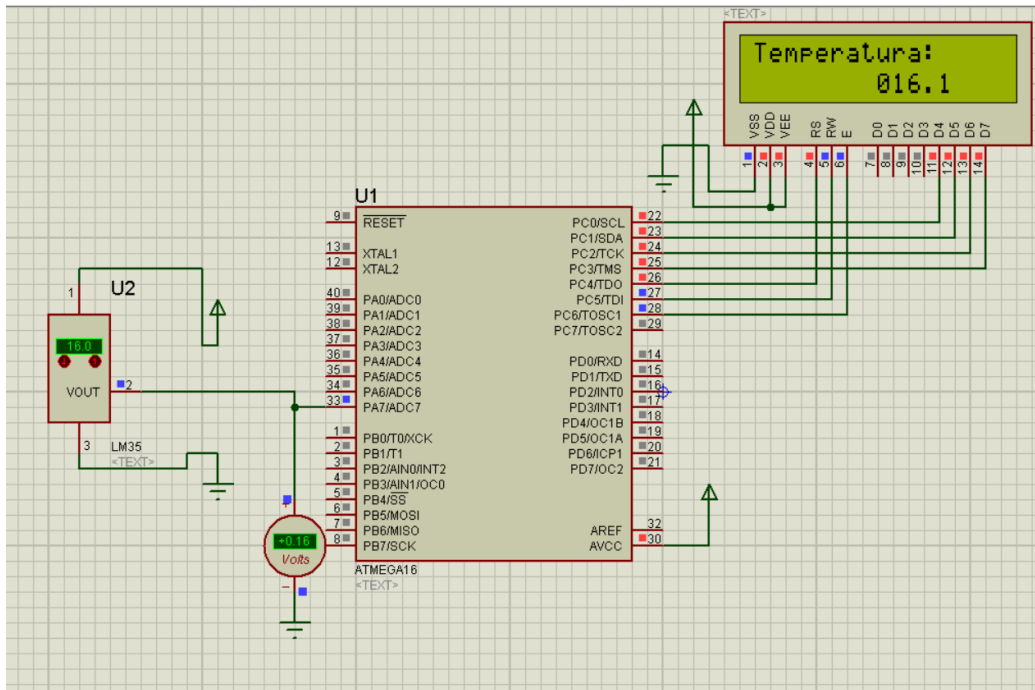
    uint8_t cero_en_bit(volatile uint8_t *LUGAR, uint8_t BIT)
    {
        return (!(LUGAR&(1<<BIT)));
    }

    uint8_t uno_en_bit(volatile uint8_t *LUGAR, uint8_t BIT)
    {
        return (LUGAR&(1<<BIT));
    }
}

```

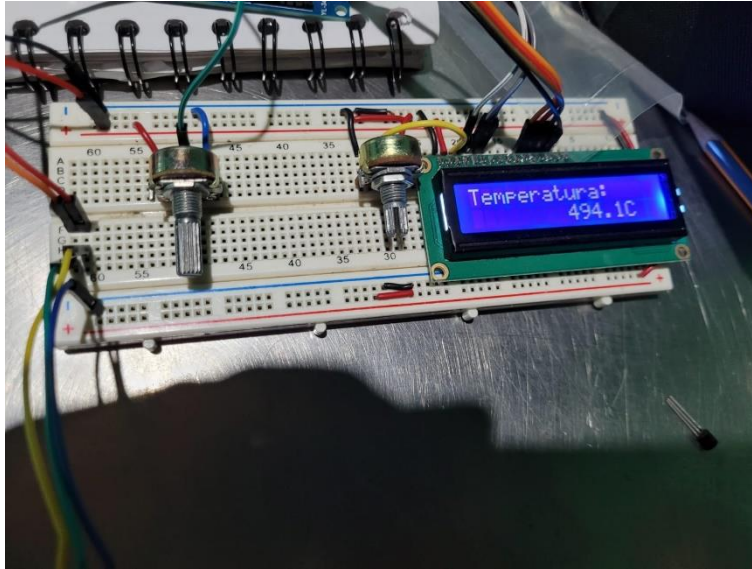
(Checar comentarios del código)

Simulacion:

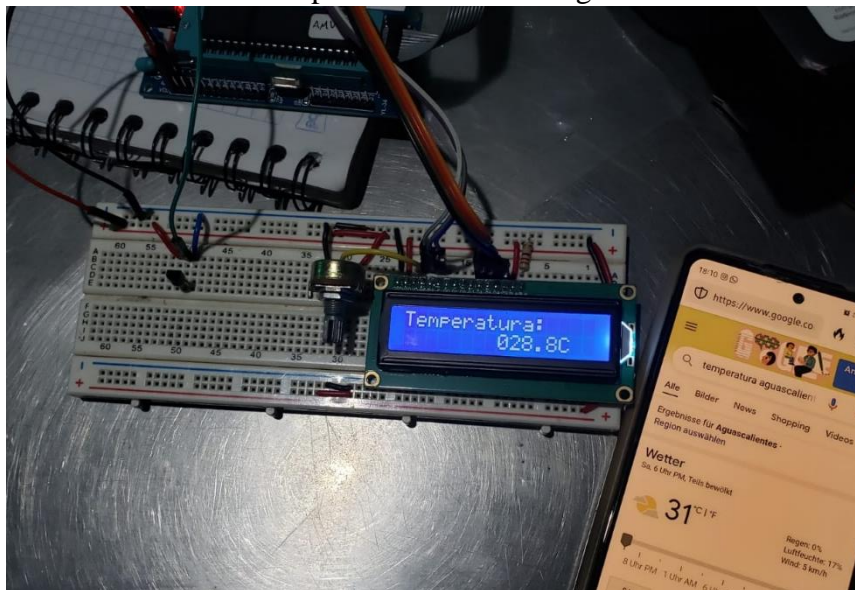


Como Podemos observar, las mediciones se hacen de manera correcta.

Implementación:



Primero se conectó el circuito anteriormente mostrado (en la sección de Simulación) donde se aseguro de que el código si hiciera la lectura de forma adecuada, en la imagen anterior podemos observar que, si mide bien la lectura de voltaje proporcionada por el potenciómetro (una prueba antes del sensor), pero falla un poco la precisión debido a que el lcd demanda energía, pero la toma de forma correcta, ya que se puso el potenciómetro para que diera los 5 volts, es por ello que muestra como si fuese una temperatura de casi 500 grados.



En este caso y por mera suerte (debido a que a varios de mis compañeros de clase les daba cosas raras), el transistor jalo de la forma adecuada ya que arrojó una medición de 28.8 grados cuando en la ambiental aparece de 31 grados, esto se debe a que se realizó la medición en un lugar mas fresco (dentro de una sala).

Prueba y analisis:

Al Momento de realizar la medición de la salida del sensor con un multímetro, como que hacia ruido o algo raro ya que de repente la medición se iba a 0, no puedo explicar por qué paso esto, ya que, dejando de medir, la lectura funciono de forma normal

Otro hecho curioso que observe durante el experimento fue que le pasamos un encendedor directamente al transistor, pero este en vez de subir la temperatura, está bajó.

Tampoco se como el porque paso esto, ya que en teoría si estaba funcionando de forma adecuada, tal vez el sensor venia defectuoso o no se , ya que incluyendo en la datasheet decía que no requería de calibración adicional.

Otra cosa que se me viene a la mente es que puede que de mediciones medio raras debido a la intervención del lcd, ya que si cambiábamos la tierra de lugar (de la parte superior hacia la parte inferior) daba una lectura ligeramente inferior de la lectura.

Conclusión:

Los sensores de este tipo, al no requerir algún tipo de calibración adicional se puede decir que son un poco impredecibles, debido a que están sujetos a una gran cantidad de factores como lo son:

- Componentes externos
- Fallas del fabricante
- Ruido del ambiente
- Entre muchas otras cosas.

Es por ello que este tipo de experimentos nos ayudan a ver alternativas para que un sensor de cierta forma “económico” (comparado con otros sensores de temperatura) pueda ayudarnos a realizar la tarea especificada.