



PROYECTO 2DO PARCIAL

## **TERMOMETRO DIGITAL**

MICROCONTROLADORES

**Uriel Everardo Sandoval Almanza**

**Aaron Medrano Márquez**

**Sergio Uriel Pérez**

ING. ROBÓTICA

Prof. Alfredo Rentería Villanueva

MAYO 2025

# TERMOMETRO DIGITAL

## OBJETIVO

Diseñar, fabricar y programar un termómetro digital que despliegue temperatura en grados Celsius y Fahrenheit, utilizando el Launchpad MSP430F5529.

## OBJETIVOS ESPECIFICOS

- Utilizar 4 displays de 7 segmentos para mostrar la temperatura y la unidad de medida (también se puede utilizar un módulo que contenga los 4 displays en un solo encapsulado)
- Los tres primeros displays de la izquierda muestran temperatura (ej. 099)
- El último display (hasta la derecha) muestra la unidad de medida (ej. C)
- Rango de medición: 0°C a 150°C
- Utilizar un botón para cambiar de unidad de medida
- Utilizar el sensor de temperatura LM35
- Diseñar y fabricar una PCB que pueda montarse directamente sobre los pines hembra del Launchpad y que contenga toda la circuitería para el/los displays, el LM35 y el botón.

## INTRODUCCIÓN

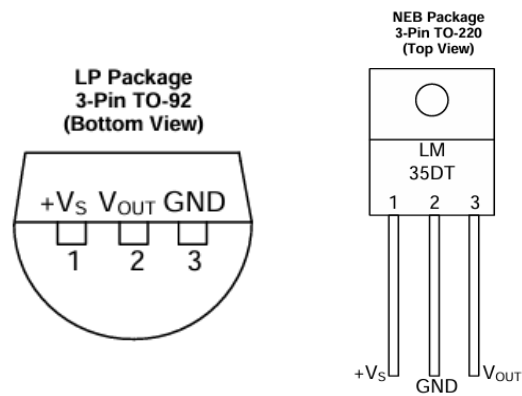
Este proyecto tiene como objetivo desarrollar un dispositivo capaz de medir la temperatura ambiente y mostrarla de forma digital en 4 display; además, se incluirá la opción de cambiar la unidad de medida entre Celsius y Fahrenheit.

El enfoque es realizar una PCB que funcione como shield para una launchpad MSP430.

## INVESTIGACIÓN

La serie LM35 son dispositivos de temperatura de circuito integrado de alta precisión, con una salida de voltaje linealmente proporcional a la temperatura en grados Celsius, además no requiere calibración ni ajuste externo para ofrecer precisiones típicas a temperatura ambiente

Está diseñado para funcionar en un rango de temperatura de  $-55^{\circ}\text{C}$  a  $150^{\circ}\text{C}$ , mientras que el modelo



1.0 Diagrama de conexión sensor LM35

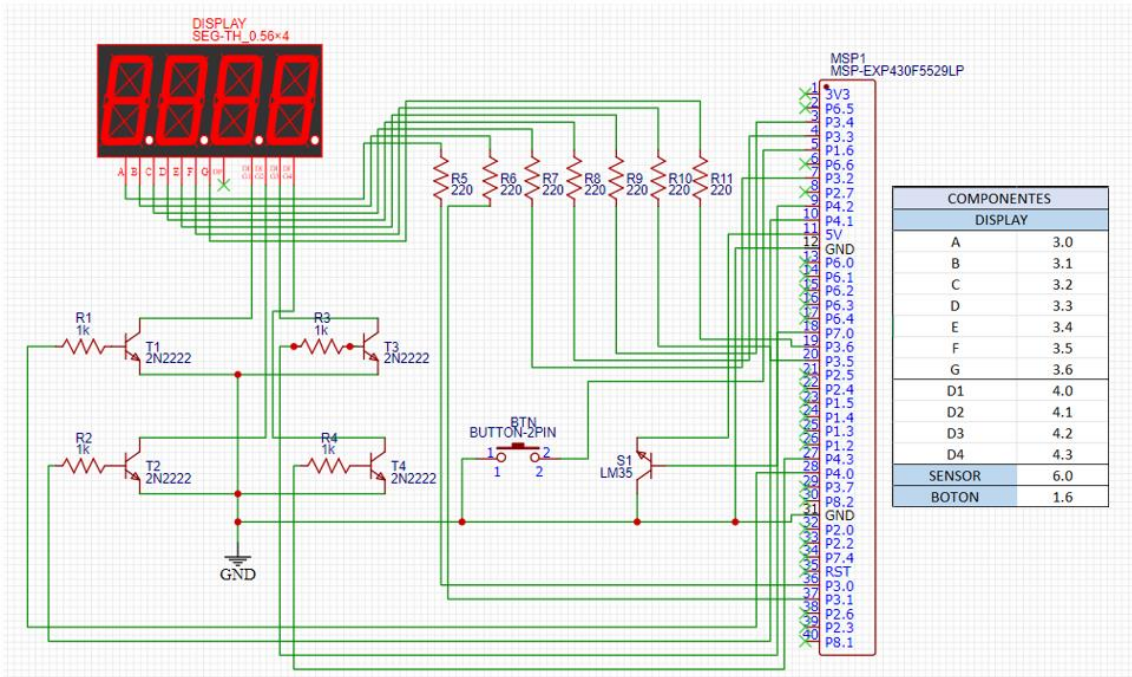
		MIN	MAX	UNIT
Supply voltage		-0.2	35	V
Output voltage		-1	6	V
Output current			10	mA
Maximum Junction Temperature, $T_{jmax}$			150	$^{\circ}\text{C}$
Storage Temperature, $T_{stg}$	TO-CAN, TO-92 Package	-60	150	$^{\circ}\text{C}$
	TO-220, SOIC Package	-65	150	

1.1 Tabla de rangos de voltaje para temperatura

Los dispositivos de la serie LM35 están disponibles en encapsulados herméticos tipo transistor TO.

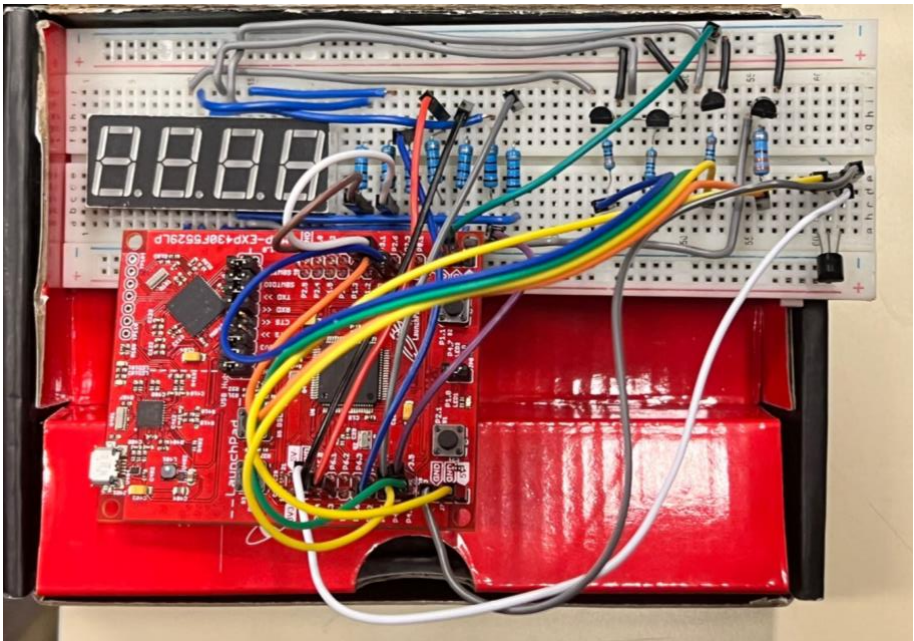
DIAGRAMAS

CONEXIONES



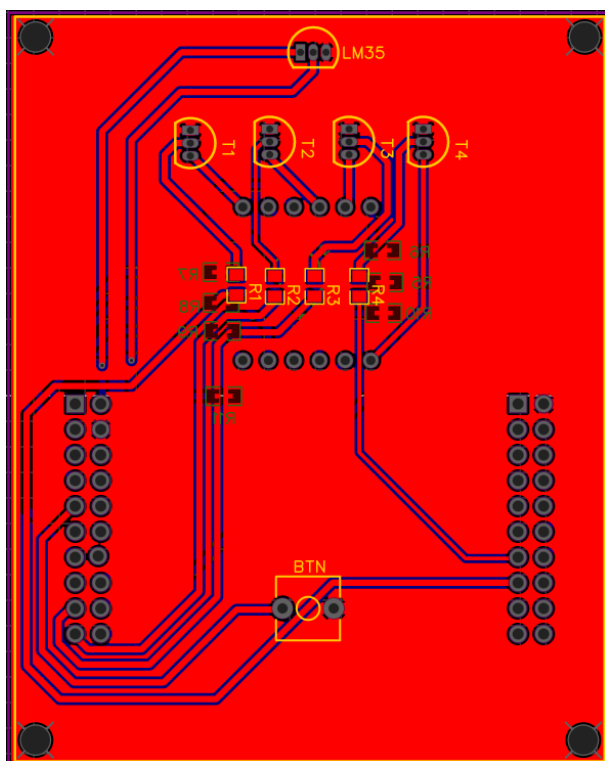
2.0 Diagrama de conexiones

PROTOTIPO

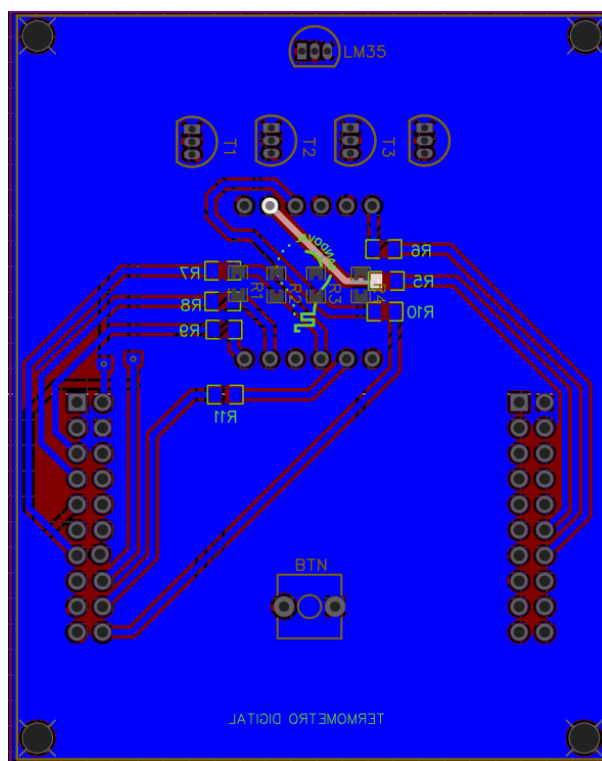


2.1 Prototipo final

## ESQUEMATICO



2.2.1 Cara frontal esquemático



2.2.2 Cara trasera esquemático

## CIRCUITO



2.3.1 Cara frontal placa final



2.3.2 Cara trasera placa final



## CODIGO

```

#include "msp430f5529.h"
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h> // para abs()

volatile uint8_t mostrarFahrenheit = 0;
unsigned char digito_actual = 0;
unsigned char letra_actual = 0x39;
unsigned char display[3] = {0, 0, 0};
const unsigned char tabla_segmentos[10]
= {
    0x3F, 0x06, 0x5B, 0x4F, 0x66,
    0x6D, 0x7D, 0x07, 0x7F, 0x6F
};

void configurarADC(void);
uint16_t leerADC(void);
uint16_t celsiusAFahrenheit(uint16_t
celsius);
void configurarBoton(void);
void configurarUART(void);
void enviarCadena(char *str);
void actualizarDisplay(uint16_t
temperatura);
void main(void) {
    WDTCTL = WDTPW | WDTHOLD;

    P3DIR |= 0x7F;
    P3OUT &= ~0x7F;
    P4DIR |= 0x0F;
    P4OUT &= ~0x0F;

    configurarADC();
    configurarBoton();
    configurarUART();

    TA0CTL0 = CCIE;
    TA0CCR0 = 1000 - 1;
    TA0CTL = TASSEL_2 | MC_1 | ID_3 |
TACLK;
    __enable_interrupt();
    __bis_SR_register(LPM0_bits);
}

void configurarADC(void) {
    P7SEL |= BIT0; // Activación
de función ADC en pin P7.0

    ADC12CTL0 = ADC12SHT0_2 |
ADC12ON;
    ADC12CTL1 = ADC12SHP;
    ADC12MCTL0 = ADC12INCH_12; //
Canal A12 (P7.0)
    ADC12CTL0 |= ADC12ENC;

```

```

}

uint16_t leerADC(void) {
    ADC12CTL0 |= ADC12SC;

    while (ADC12CTL1 & ADC12BUSY);

    return ADC12MEM0;
}

uint16_t celsiusAFahrenheit(uint16_t
celsius) {
    return (celsius * 9 / 5) + 32;
}

void configurarBoton(void) {
    P1DIR &= ~BIT6;

    P1REN |= BIT6;

    P1OUT |= BIT6;

    P1IE |= BIT6;

    P1IES |= BIT6;

    P1IFG &= ~BIT6;
}

void configurarUART(void) {
    P4SEL |= BIT4 + BIT5;

    UCA1CTL1 |= UCSWRST;

    UCA1CTL1 |= UCSSEL_2;

    UCA1BR0 = 6;

    UCA1BR1 = 0;

    UCA1MCTL = UCBRS_0 + UCBRF_13 +
UCOS16;

    UCA1CTL1 &= ~UCSWRST;

```

```

}

void enviarCadena(char *str) {
    while (*str != 0) {
        while (!(UCA1IFG & UCTXIFG));

        UCA1TXBUF = *str++;
    }
}

void actualizarDisplay(uint16_t
temperatura) {
    display[0] = temperatura / 100;

    display[1] = (temperatura / 10) % 10;

    display[2] = temperatura % 10;
}

#pragma vector = TIMER0_A0_VECTOR
__interrupt void TIMER0_A0_ISR(void) {
    static uint16_t sendCounter = 0;

    static unsigned long adc_accum = 0;

    static uint16_t sample_count = 0;

    static uint16_t last_temp = 0;

    uint16_t adc_value, mv, temp_c,
temp_mostrar;

    char buffer[32];

    adc_value = leerADC();

    adc_accum += adc_value;

    sample_count++;

```

```

    if (sample_count >= 4) { // promedio
de solo 4 muestras

    adc_value = adc_accum / 4;

    adc_accum = 0;

    sample_count = 0;

    mv = (uint32_t)adc_value * 3300 /
4095;

    temp_c = mv / 10;

    if (abs(temp_c - last_temp) >= 1) { //
cambio de al menos 1 °C

        last_temp = temp_c;

    }

    temp_mostrar = mostrarFahrenheit ?
celsiusAFahrenheit(last_temp) :
last_temp;

    if (temp_mostrar > 999)
temp_mostrar = 999;

    actualizarDisplay(temp_mostrar);

    letra_actual = mostrarFahrenheit ?
0x71 : 0x39;

    if (++sendCounter >= 500) {

        sendCounter = 0;

        sprintf(buffer, "%d %c\r\n",
temp_mostrar, mostrarFahrenheit ? 'F' :
'C');

        enviarCadena(buffer);

    }

}

```

```

P4OUT &= ~0x0F;

switch (digito_actual) {

    case 0:

    case 1:

    case 2:

        P3OUT =
tabla_segmentos[display[digito_actual]];

        P4OUT |= (1 << digito_actual);

        break;

    case 3:

        P3OUT = letra_actual;

        P4OUT |= BIT3;

        break;

}

digito_actual++;

if (digito_actual > 3)

    digito_actual = 0;

}

#pragma vector = PORT1_VECTOR
__interrupt void Port_1(void) {

    if (P1IFG & BIT6) {

        mostrarFahrenheit ^= 1;

        P1IFG &= ~BIT6;

    }

}

```



## BIBLIOGRAFIA

Texas Instruments. (s.f.). LM35 precision centigrade temperature sensors (DS00611B).

Recuperado de <https://www.ti.com/lit/ds/symlink/lm35.pdf?ts=1736143362540>

## CONCLUSIÓN

Este proyecto permitió desarrollar un sistema sencillo y funcional para medir y mostrar la temperatura ambiente en tiempo real, utilizando 1 display de 4 dígitos 7 segmentos y un sensor LM35. La incorporación de un botón para cambiar entre Celsius y Fahrenheit mejora su utilidad, además, el diseño de una PCB tipo shield para el Launchpad MSP430 hace que la solución sea compacta, ordenada y fácil de integrar en futuras aplicaciones.