

## Laboratorio 6

Termómetro Digital

Josué Meneses Díaz

14 de agosto de 2024

# **Objetivos**

- Identificar si un termistor es del tipo NTC o PTC.
- Utilizar los pines analógicos del MCU para la lectura de datos.
- Utilizar los pines digitales del MCU como entradas y salidas digitales.
- · Realizar operaciones matemáticas con el MCU.

### 1 Materiales

- 1 x Arduino UNO.
- 2 x Display de 7 segmentos de cátodo común.
- 2x 2N2222 [1]
- 7 x resistencia 330Ω (referencial).
- 2 x 4k7Ω.
- 2 x 10kΩ.
- · 1 x Termistor NTC o PTC.
- 1 x multitester.
- 1 x Termómetro

## 2 Indicaciones

En este laboratorio ampliaremos el circuito realizado previamente. Modificaremos su uso para que los displays muestren un valor de temperatura. La temperatura será medida utilizando un termistor conectado a una de las entradas analógicas del MCU. Además, utilizaremos un pulsador para *mantener* el valor de temperatura mostrada por los displays hasta que el pulsador sea nuevamente accionado. En la Figura 1 se muestra el circuito esquemático del nuevo circuito.

Una vez terminadas las actividades de esta guía **no desarme su circuito**. Este será utilizado en los próximos laboratorios.

## 3 Procedimiento

- Modifique el programa desarrollado en el laboratorio anterior para mostrar los valores medidos en la entrada analógica A0 de su placa de desarrollo Arduino.
- Utilice las siguientes conexiones para el circuito (ver Figura 1):



```
- Pin D2: Circuito Pulsador.
```

- Pin D5: Transistor Q1.
- Pin D6: Transistor Q2.
- Pin D7: segmento a.
- Pin D8: segmento b.
- Pin D9: segmento c.
- Pin D10: segmento d.
- Pin D11: segmento e.
- Pin D12: segmento f.
- Pin D13: segmento g.
- Pin A0: Circuito termistor.
- Tanto la alimentación de +5V, como la de referencia GND serán suministradas mediante los pin 5V y GND de la placa arduino.
- Note que las conexiones del circuito son iguales a las del laboratorio anterior, solo hemos agregado un pulsador y un termistor.

Para medir valores lógicos con los pines digitales de la placa utilizamos la función digitalRead(). El siguiente código es un ejemplo de como configurar y leer valores con el pin digital D2:

```
int val = 0;  // variable de almacenamiento del valor medido

void setup() {
   pinMode( 2, INPUT);  // Establece el pin D2 como entrada.
}

void loop() {
   ...
   val = digitalRead(2);  // Almacena en val el valor del pin D2
   ...
}
```

• Finalmente, para medir los valores del termistor, leemos el voltaje del divisor de tensión mediante la entrada analógica A0 utilizando la función analogRead():

```
int val = 0; // variable de almacenamiento para el termistor

void setup() {
    void loop() {
        ...
        val = analogRead(AO); // Almacena en val la lectura de la entrada analógica AO
        ...
}
```

#### 3.1 Calibración de termistor

Los valores medidos por el conversor análogo-digital (ADC) del MCU abarcan el rango de números enteros (int) entre 0 a 1023 (2<sup>10</sup>), donde 1023 corresponde al voltaje de referencia de +5V y 0 al GND. Por lo tanto, es necesario calibrar la resistencia del termistor con respecto a la temperatura y relacionar el voltaje medido sobre el divisor de tensión, donde se encuentra el termistor (ver Figura 1).

La calibración del termistor será realizada mediante el método de comparación:



- Utilizando un multitester y un termómetro, someta a diferentes temperaturas el termistor a calibrar, registrando para cada temperatura la resistencia medida.
- Realice la medición anterior por lo menos para tres temperaturas distintas.
- Encuentre el mejor ajuste utilizando un la ecuación:

$$R(\Omega) = A \cdot e^{B \cdot T(\cdot C)}$$

- Utilice la ecuación encontrada para realizar la conversión de la lectura del termistor a temperatura (Voltaje medidio por el ADC). Utilice la función exp() o log() [2] para calcular la temperatura leida.
- Con los valores encontrados, determine que tipo de termistos utilizando.

#### 3.2 Interrupciones - 1

Agregaremos un botón pulsador a nuestro termómetro digital que permita mantener fija la temperatura mostrada en el display hasta que el pulsador vuelva a ser presionar. Para ello:

- Utilice el pin digital D2 para conectar el pulsador.
- Modifique su programa para que, si el pulsador es presionado, mantenga el valor actual de temperatura mostrada hasta volver a presionar el pulsador.
- Note que el pin D2 posee una resistencia *Pull-up* (ver Figura 1) y que al accionar el pulsador el pin queda conectado a tierra.

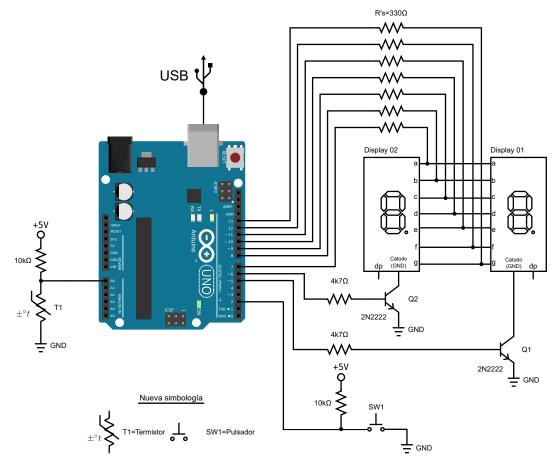


Figura 1: Termómetro digital con Arduino UNO y D7S

#### Electronica Digital y Microcontroladores Universidad de Santiago de Chile Departamento de Física - Ingeniería Física

Lab. 6 - Termómetro Digital con D7S

### 4 Preguntas reporte

- 1. En una tabla y gráfico presente los datos obtenidos para la resistencia y temperatura del termistor, junto con la línea de tendencia encontrada. Agregue los cálculos realizados para convertir el voltaje medido en temperatura. (1 punto)
- 2. En base a sus mediciones, ¿Qué tipo de termistor se ha utilizado? ¿Cómo ha llegado a esa conclusión? **(1 punto)**
- 3. Agregue un diagrama de flujo del nuevo programa. Explique la filosofía de su programa. Adjunte el programa .ino (Arduino [3]) y la simulación .sim1 (SimulIde [4]) desarrollada. (1 punto)
- 4. ¿Qué problemas detecta en el uso del pulsador conectado en el pin digital D2? (1 punto)
- 5. Calcule la **resolución** del ADC de Arduino y su **paso de voltaje. (1 punto)**
- 6. Para 25°C, ¿Cuánto cambia la lectura de la temperatura si aumenta/disminuye la medición de MCU en un **paso de voltaje**? Explique la diferencia. **(0.5 puntos)**
- 7. ¿En qué rango de temperatura posee una mejor **resolución** el termistor para la configuración ensayada? Justifique su respuesta estimando la resolución para dos temperaturas extremas. ¿Modificaría en algo el circuito utilizado? **(0.5 puntos)**

### Referencias

- [1] O. Semiconductor, «P2N2222A Amplifier Transistors NPN Silicon». [En línea]. Disponible en: https://www.onsemi.com/pdf/datasheet/p2n2222a-d.pdf
- [2] A. Libc, «Avr-Libc: <math.H>: Mathematics». Accedido: 26 de junio de 2022. [En línea]. Disponible en: https://www.nongnu.org/avr-libc/user-manual/group\_\_avr\_\_math.html#ga9bf5d952c5c93c70f9e66c9794d406c9
- [3] Arduino, «Arduino IDE». Accedido: 21 de junio de 2022. [En línea]. Disponible en: https://www.arduino.cc/en/software
- [4] SimulIDE, «Home Page SimulIDE». Accedido: 21 de junio de 2022. [En línea]. Disponible en: https://www.simulide.com/p/home.html