



Instituto Politécnico Nacional
Escuela Superior de Cómputo
Departamento de Ciencias e Ingeniería
de la Computación



Sistemas en Chip

Práctica 16

“Termómetro de 0 a 50 °C (32 a 122 °F)”

Profesor: Fernando Aguilar Sánchez

Grupo: 6CM1

Equipo 3

Alumnos:

Ocampo Téllez Rodolfo

Patlani Mauricio Adriana

Ruvalcaba Flores Martha Catalina

Sandoval Hernández Eduardo

Fecha de entrega: 15/01/2023

Objetivo

Al término de la sesión, los integrantes del equipo contarán con la habilidad de hacer uso del convertidor analógico digital del microcontrolador implementando un termómetro de 0 a 50 °C (32 a 122°F).

Introducción Teórica

El microcontrolador ATMEGA8535 fabricante ATMEL. En la figura 1 se muestra el microcontrolador ATMEGA8535 la cual maneja datos de 8 bits es decir su bus de datos de 8 bits. Aunque este contiene tres registros los cuales son el x, y y z, los cuales manejan datos de 16 bits.

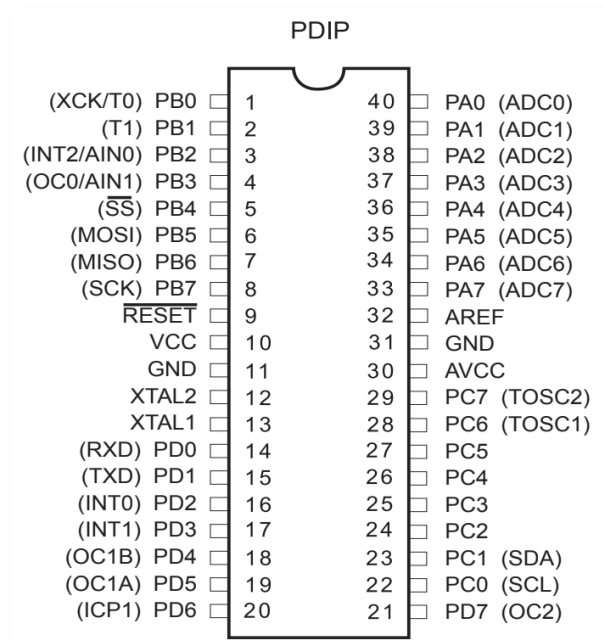


Figura 1. Configuración de pines ATmega8535.

La comunicación interna del microcontrolador se categoriza en 4 puertos, en la figura 1 se puede analizar la etiquetación de los puertos PA0 al PA7, PB0 al PB7, PC0 al PC7 y PD0 al PD7. Cabe recordar que maneja datos de 8 bits.

Posee un oscilador interno de 1MHz, sin embargo, como es un oscilador RC, es susceptible a variar la frecuencia con respecto a las variaciones de temperatura. Por otro lado, puede conectarse un oscilador de 0 a 8 MHz o de 0 a 16 MHz.

El microcontrolador se alimenta de las terminales 10 y 11 como lo muestra la figura 1, los cuales son el VCC (5 V y una tolerancia de $\pm 0.5V$) y GND. Sin embargo, el convertidor se alimenta de forma externa en la terminal 31 y 32, los cuales son GND y AREF.

Tiene 8 canales que sirven a un ADC de 10 bits, los 8 canales son de una sola terminal, 7 canales son diferenciales para únicamente paquetes TQFP y 2 de esos canales diferenciales permiten ganancia programable de 1x, 10x, o 200x.

Conversión Analógica-Digital

La conversión analógica-digital consiste en la transcripción de señales analógicas en señal digital, con el propósito de facilitar su procesamiento (codificación, comprensión, etcétera) y hacer la señal resultante (digital) más inmune al ruido y otras interferencias a las que son más sensibles las señales analógicas.

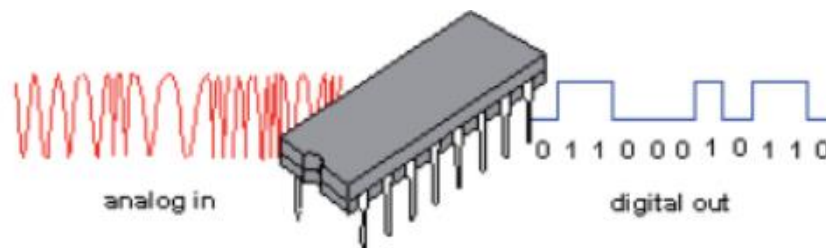


Figura 2. Conversión de información analógica a una digital.

Para realizar esa tarea, se utiliza un ADC (Analog-to-Digital Converter – Conversor Analógico-Digital) que tiene que efectuar los siguientes procesos:

Muestreo de la señal analógica, que es la medición periódica de la amplitud de una señal para poder evaluar su nivel. Sin embargo, este proceso no se contempla desde el punto de vista matemático, puesto que es un recurso técnico sin modelo matemático y con limitaciones prácticas. Durante el muestreo, la señal sigue siendo analógica ya que ésta puede tomar cualquier valor.

Cuantización o cuantificación, en este proceso lo que se mide es el voltaje de cada muestra, asignándoles un margen de valor de una señal analizada a un único nivel de salida. La desventaja es que siempre el resultado lleva una distorsión llamada “ruido de cuantificación”.

Codificación del resultado, codificar es la acción por la cual los valores que se obtuvieron en la cuantificación se traducen en un código binario o en otro tipo de códigos similares. La longitud de la palabra binaria obtenida depende de la resolución (en bits) que se decida utilizar.

LM35

Los sensores son dispositivos ampliamente usados en multitud de circuitos. Los hay de temperatura, de humedad, de humo, de luz, y un largo etc. Son elementos que permiten medir alguna magnitud y transformarla en una respuesta de tensión. La señal analógica de salida se puede transformar a digital de forma sencilla y así poder usar este tipo de sensores con circuitos digitales, pantallas LCD, una placa Arduino, etc.

El LM35 es uno de los sensores más populares y utilizados de todos, ya que es un circuito electrónico sensor que puede medir temperatura. Su salida es analógica, es decir, te proporciona un voltaje proporcional a la temperatura. El sensor tiene un rango desde -55°C a 150°C . Su popularidad se debe a la facilidad con la que se puede medir la temperatura. Incluso no es necesario de un microprocesador o microcontrolador para medir la temperatura. Dado que el sensor LM35 es analógico, basta con medir con un multímetro, el voltaje a salida del sensor.

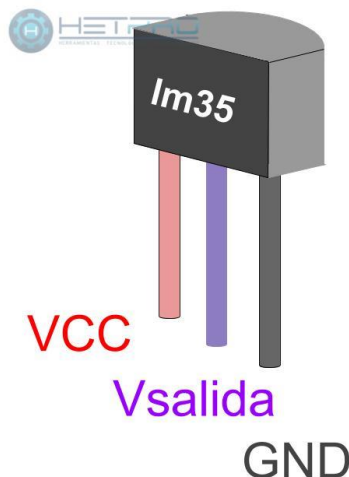


Figura 3.. Sensor LM35 en empaquetado TO-92.

Para convertir el voltaje a la temperatura, el LM35 proporciona 10mV por cada grado centígrado. También cabe señalar que ese sensor se puede usar sin offset, es decir que si medimos 20mV a la salida, estaremos midiendo 2°C. Sus características principales son:

- Resolución: 10mV por cada grado centígrado.
- Voltaje de alimentación. Por ejemplo, este sensor se puede alimentar desde 4Vdc hasta 20Vdc.
- Tipo de medición. Salida analógica.
- Numero de pines: 3 pines, GND, VCC y VSalida.
- No requiere calibración.
- Tiene una precisión de $\pm 1/4^{\circ}\text{C}$.
- Esta calibrado para medir $^{\circ}\text{C}$.
- Consumo de corriente: 60 μA

Materiales y Equipo empleado

- CodeVision AVR
- AVR Studio 4
- Microcontrolador ATmega 8535
- 3 Display Cátodo común
- 8 Resistores de 330 Ω a $1/4$ W
- 3 Resistores de 1K Ω a $1/4$ W
- 3 Transistores BC547 o 2N2222

Desarrollo Experimental

1.- Diseñe junto con el siguiente circuito, un termómetro que trabaje en el rango de 0 a 50 $^{\circ}\text{C}$ (32 a 122°F).

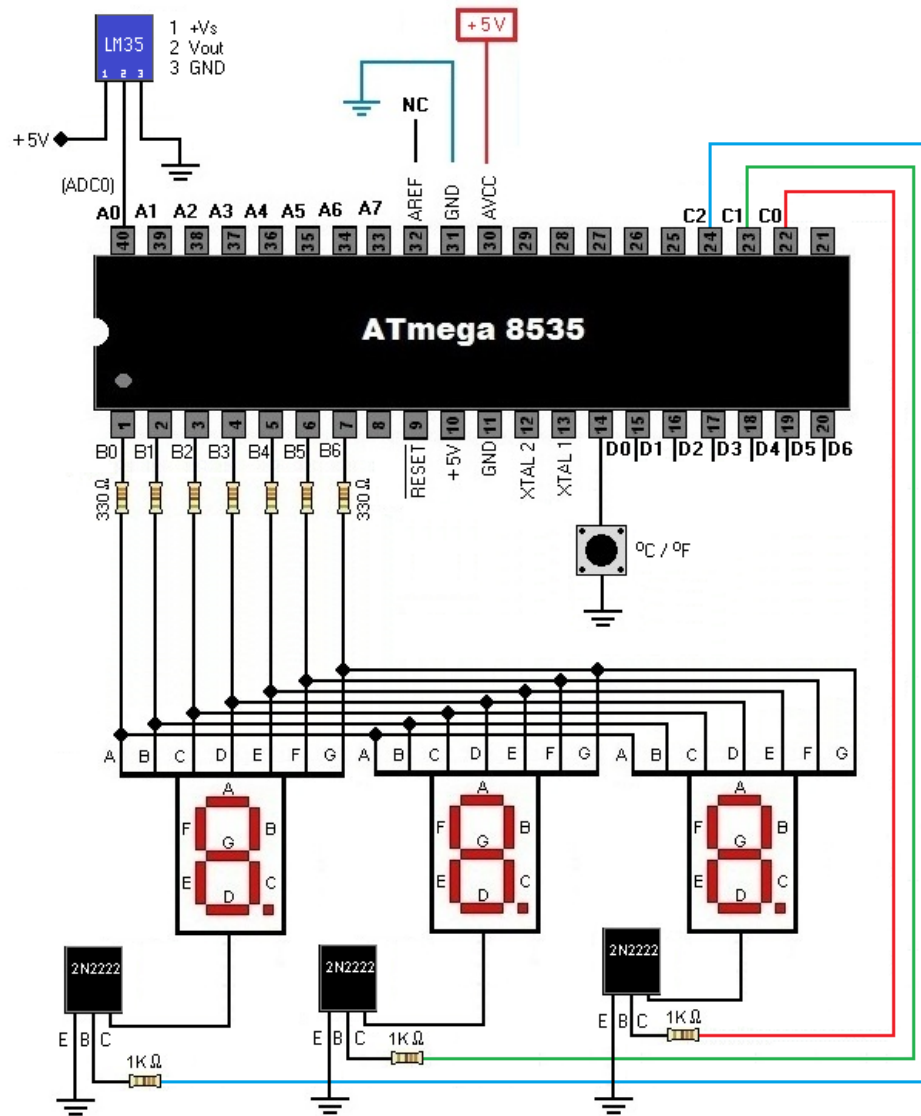


Figura 4. Circuito para el Termómetro.

Estructura del programa

Código de la configuración de los periféricos utilizados y código del programa principal en C, proporcionados por el IDE de CodeVision.

```
#include <mega8535.h>
#include <delay.h>
#define boton PIND.0

bit botonp;
bit botona;
bit grado = 0;
```

```
const char tabla7segmentos [10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x6f};
unsigned char valorADC;
unsigned char valorFinal;
int auxValor1, auxValor2;
#define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) | (1<<ADLAR))

unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | ADC_VREF_TYPE;
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=(1<<ADSC);
    // Wait for the AD conversion to complete
    while ((ADCSRA & (1<<ADIF))==0);
    ADCSRA|=(1<<ADIF);
    return ADCH;
}

void main(void)
{
    // Input/Output Ports initialization
    // Port A initialization
    DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1)
    | (0<<DDA0);
    PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2)
    | (0<<PORTA1) | (0<<PORTA0);
    // Port B initialization
    DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1)
    | (1<<DDB0);
    PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2)
    | (0<<PORTB1) | (0<<PORTB0);
    // Port C initialization
    DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) | (1<<DDC2) | (1<<DDC1)
    | (1<<DDC0);
    PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2)
    | (0<<PORTC1) | (0<<PORTC0);
    // Port D initialization
    DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1)
    | (0<<DDD0);
    PORTD=(1<<PORTD7) | (1<<PORTD6) | (1<<PORTD5) | (1<<PORTD4) | (1<<PORTD3) | (1<<PORTD2)
    | (1<<PORTD1) | (1<<PORTD0);
    ...
    ...
    ...
    while (1)
```

```
{
  if(boton == 0)
    botona = 0;
  else
    botona = 1;
  if((botonp == 1) && (botona == 0)){ //Flanco de 1 a 0
    grado = ~grado;
    delay_ms(40);
  }
  if((botonp == 0) && (botona == 1)) //Flanco de 0 a 1
    delay_ms(40);

  valorADC = 10*read_adc(0);
  if(grado){
    valorFinal = 0.196*valorADC;
    delay_ms(10);
    PORTB = tabla7segmentos [0];
    PORTC = 0x03;
    delay_ms(10);
    auxValor1 = valorFinal*0.1;
    PORTB = tabla7segmentos [auxValor1];
    PORTC = 0x05;
    delay_ms(10);
    PORTB = tabla7segmentos [valorFinal%10];
    PORTC = 0x06;
    delay_ms(10);
  }else{
    valorFinal = 1.8*(0.196*valorADC)+32;
    delay_ms(10);
    auxValor1 = valorFinal*0.01;
    PORTB = tabla7segmentos [auxValor1];
    PORTC = 0x03;
    delay_ms(10);
    auxValor1 = valorFinal - auxValor1*100;
    auxValor2 = valorFinal*0.1;
    auxValor2 = valorFinal - auxValor2*10;
    auxValor1 = (auxValor1-auxValor2)*0.1;
    PORTB = tabla7segmentos [auxValor1];
    PORTC = 0x05;
    delay_ms(10);
    PORTB = tabla7segmentos [auxValor2];
    PORTC = 0x06;
    delay_ms(10); }
  botonp = botona;
}
}
```

Simulaciones

La simulación se realizó en el software Proteus Design Suite, previamente cargado el programa en formato “.hex” obtenido del software CodeVision AVR, el cual es un software para el microcontrolador Microchip AVR y sus contrapartes XMEGA.

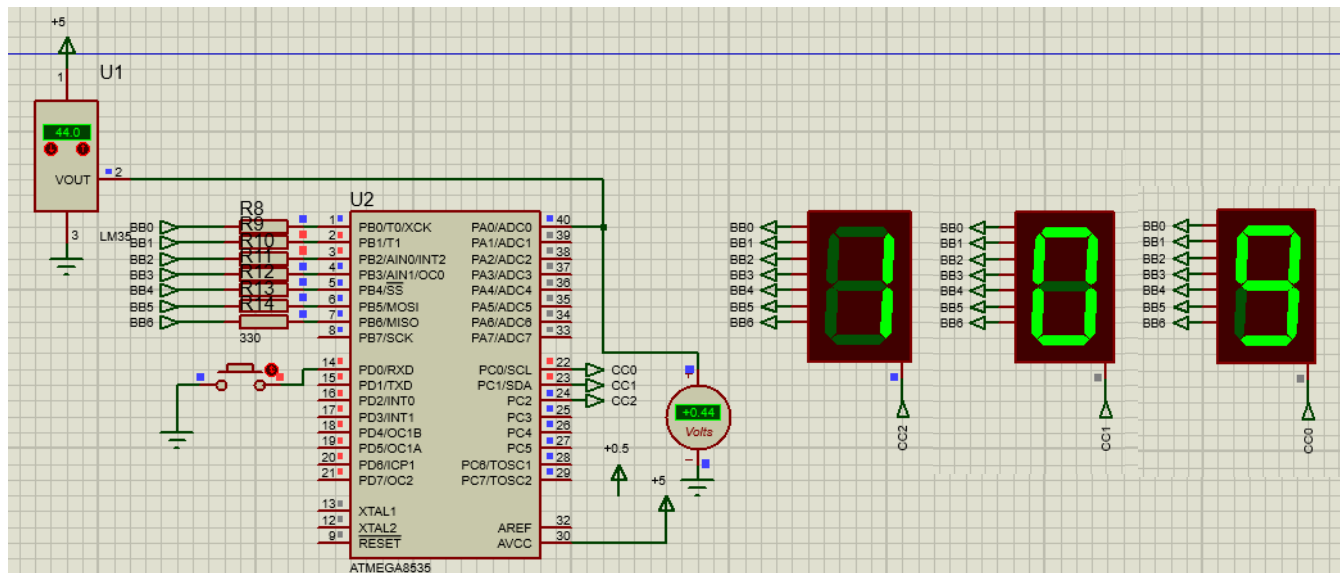


Figura 5. Simulación de la práctica 16.

Fotografía del circuito armado

A continuación, se muestra la figura 6 la evidencia sobre la realización y prueba de la practica 16.

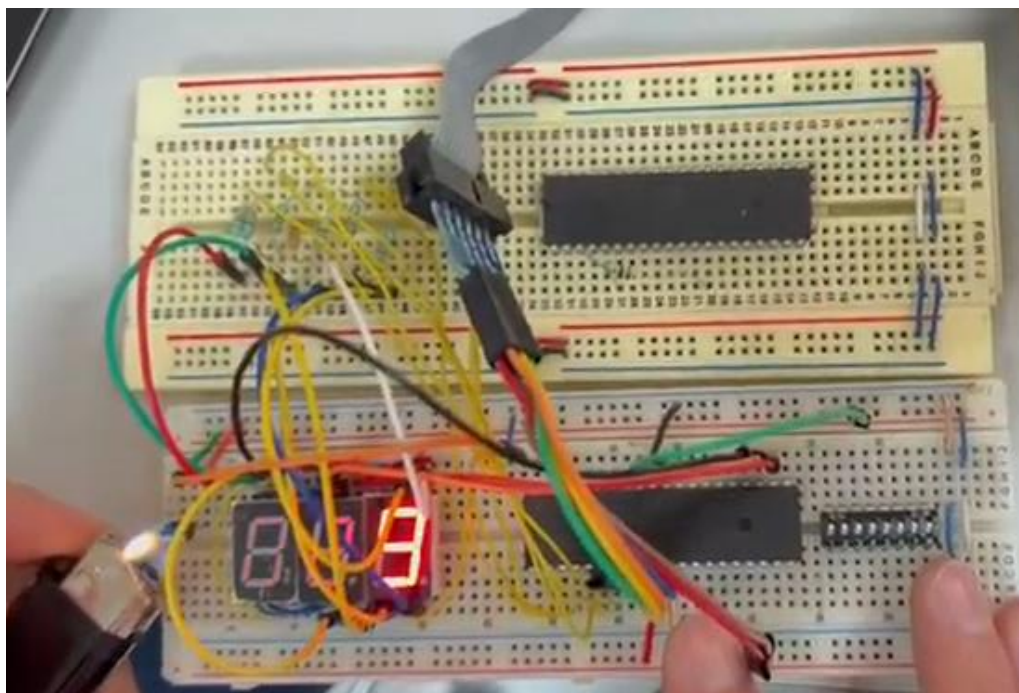


Figura 6. Circuito armado y funcionamiento de la práctica 16.

Observaciones y conclusiones Individuales

Ocampo Téllez Rodolfo

Esta práctica guardo cierta similitud con la anterior, puesto que el rango de valores a medir fue igual, de 0 a 50, aunque en esta ocasión el valor analógico leído y convertido a uno digital mediante el ADC del Atmega fue la temperatura obtenida mediante el sensor LM35. Para realizar el programa se tuvo que tomar en cuenta que el sensor utilizado incrementaba únicamente 10 mV/°C, por lo que el máximo valor obtenido apenas llegaría a 0.5 V, como el microcontrolador trabaja con 5 V se realizó la modificación en el código para que el valor entregado por `read_adc(0)` se incrementara en 10x. trabajando con ello para la obtención de los valores que debían mostrarse en los displays, y con base a ello al utilizarse 8 bits de resolución se realizaron saltos de $0.0196 \times 10 = 0.196$ para obtener todos los valores del rango de grados centígrados. En el caso de que selección para mostrar en Fahrenheit se aplicaba un procedimiento bastante similar utilizando la fórmula de conversión de °C a °F proporcionada por el profesor.

Patlani Mauricio Adriana

En esta práctica vemos otra aplicación del convertidor analógico a digital del microcontrolador, siendo muy similar a la práctica anterior solo que esta vez, la señal analógica a convertir será la temperatura. Para ello usamos una de las funcionalidades del sensor LM35 que lee cada 10mV como 1°C; sabiendo esto se armó el circuito para ver los valores en grados centígrados y grados Fahrenheit haciendo la conversión respectiva y para ello ampliando el valor máximo para que se pudiera mostrar ambos datos en los displays.

Ruvalcaba Flores Martha Catalina

A partir de los conocimientos adquiridos desde la práctica 13 sobre el convertidor analógico a digital, en esta práctica se hace uso de ello para implementar un termómetro de 0 a 50 grados Centígrados. Para ello se usa un LM35, el cual es un sensor que proporciona los valores de temperatura, asimismo fue necesario checar su datasheet para conocer el rango de temperatura y su sensor lineal para reflejar de forma correcta el valor en los displays. Para realizar la conversión de analógico a digital, se utilizó la función de `read_adc(0)` de la práctica 15, tomando los 10 bits (0 a 1023). Finalmente, para cumplir con el objetivo de la práctica, se solicitó realizar la conversión de grados Centígrados a Fahrenheit y viceversa, para ello se utilizaron las fórmulas que aparecen en el video.

Sandoval Hernández Eduardo

Esta práctica fue muy similar a la anterior tanto que incluso se reutilizó el mismo circuito adaptando lo que fuera necesario, el mayor problema fue que, aunque el código parecía sencillo de implementar en realidad se complicó debido a que para los grados Fahrenheit los datos en los displays no se mostraban de la forma adecuada, lo cual posteriormente se solucionó haciendo arreglos con las variables, el otro problema fue que el sensor LM35 no tenía una salida de voltaje lo suficientemente grande para la escala que se estaba manejando en el microcontrolador, por tanto se tuvo que amplificar ese valor, como resultado se obtuvo un sensor de temperatura que funciona de manera aceptable.

Bibliografía

- Atmel Corporation. (2006). 8-bit Microcontroller with 8K Bytes In-System Programmable Flash. [Online]. Disponible en: <https://drive.google.com/file/d/1acpQaDlsyLHr3w3ReSgrlXRbshZ5Z-lb/view>
- F. Aguilar. (Mayo, 2020). Microcontroladores Termómetro. [Online]. Disponible en: https://youtu.be/V_UfFNtFIBI
- Menna. (s.f.). Cómo funciona un convertidor analógico a digital. [Online]. Disponible en: <https://como-funciona.co/como-funciona-un-convertidor-analogico-a-digital/>
- Isaac. (Febrero. 2021). LM35: información completa sobre este sensor de temperatura. [Online]. Disponible en: <https://www.hwlibre.com/lm35/>
- R. Marmolejo. (Febr, 2022). LM35 – El sensor de temperatura más popular. [Online]. Disponible en: <https://hetpro-store.com/TUTORIALES/lm35/>