

# Leer el sensor de temperatura LM35 en Arduino

Luis del Valle Hernández

En este fragmento de código se va a mostrar como debemos **leer el sensor de temperatura LM35 en Arduino**.

En este caso hemos optado por un sensor bastante normal que detecta temperaturas desde -55°C a 150°C, 1°C equivale a 10mV y soporta voltajes de entre 4V y 30V. Todo esta información se ha obtenido de la ficha técnica del LM35 (hoja de datos).

Cuando leemos un sensor analógico con Arduino lo hacemos a través de la función `analogRead` que nos da un valor entre 0 y 1023, 1024 valores posibles.

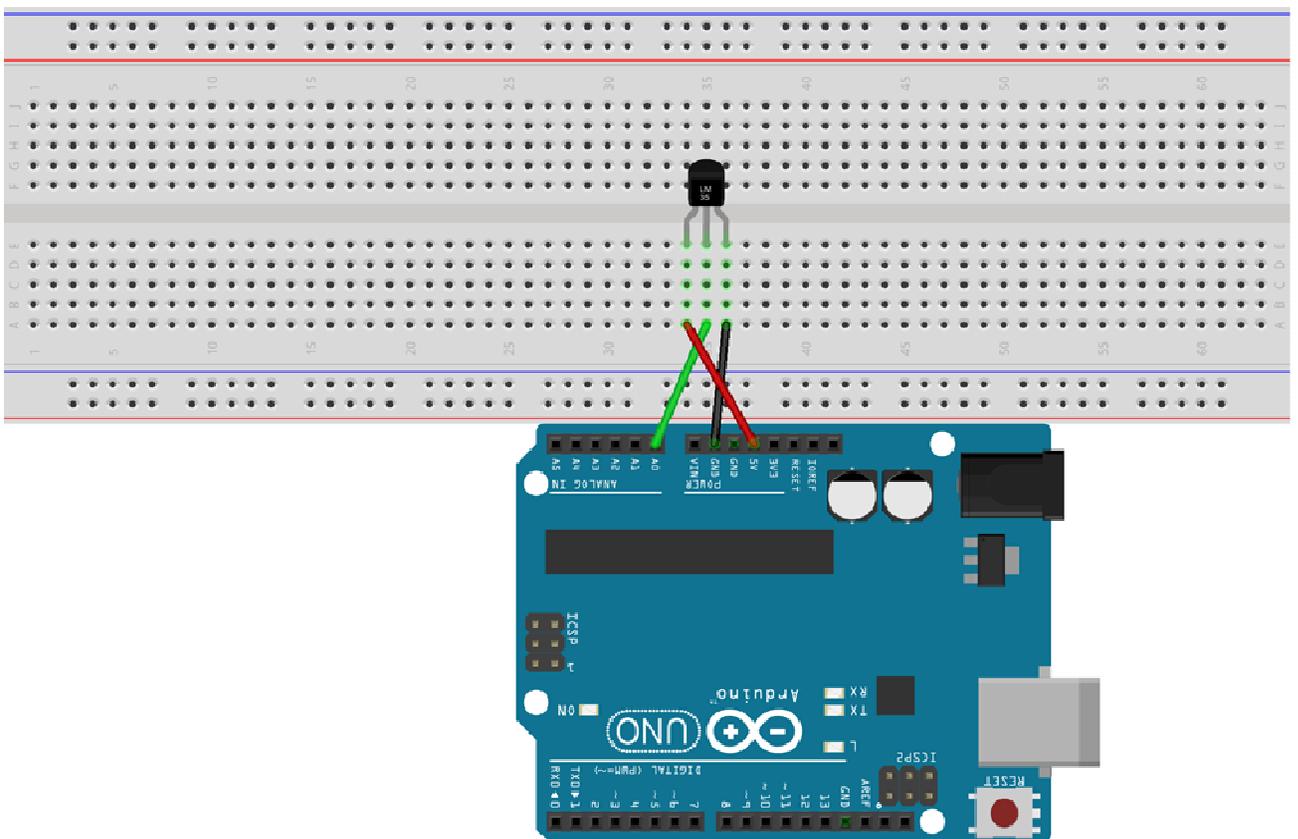
Si tenemos 0V a la entrada nos devolverá 0 y si tenemos 5V nos devolverá 1023.

A partir de esta información podemos obtener una fórmula matemática que nos calcule la temperatura en función del voltaje que nos facilita el LM35.

$$\text{Temperatura} = \text{Valor} * 5 * 100 / 1024$$

*\*Para más detalle de cómo se obtiene esta fórmula puedes leer el Apéndice.*

Lo primero es ver **cómo se conecta el LM35 con Arduino** y aquí os dejo el esquema eléctrico que vamos a seguir.



fritzing

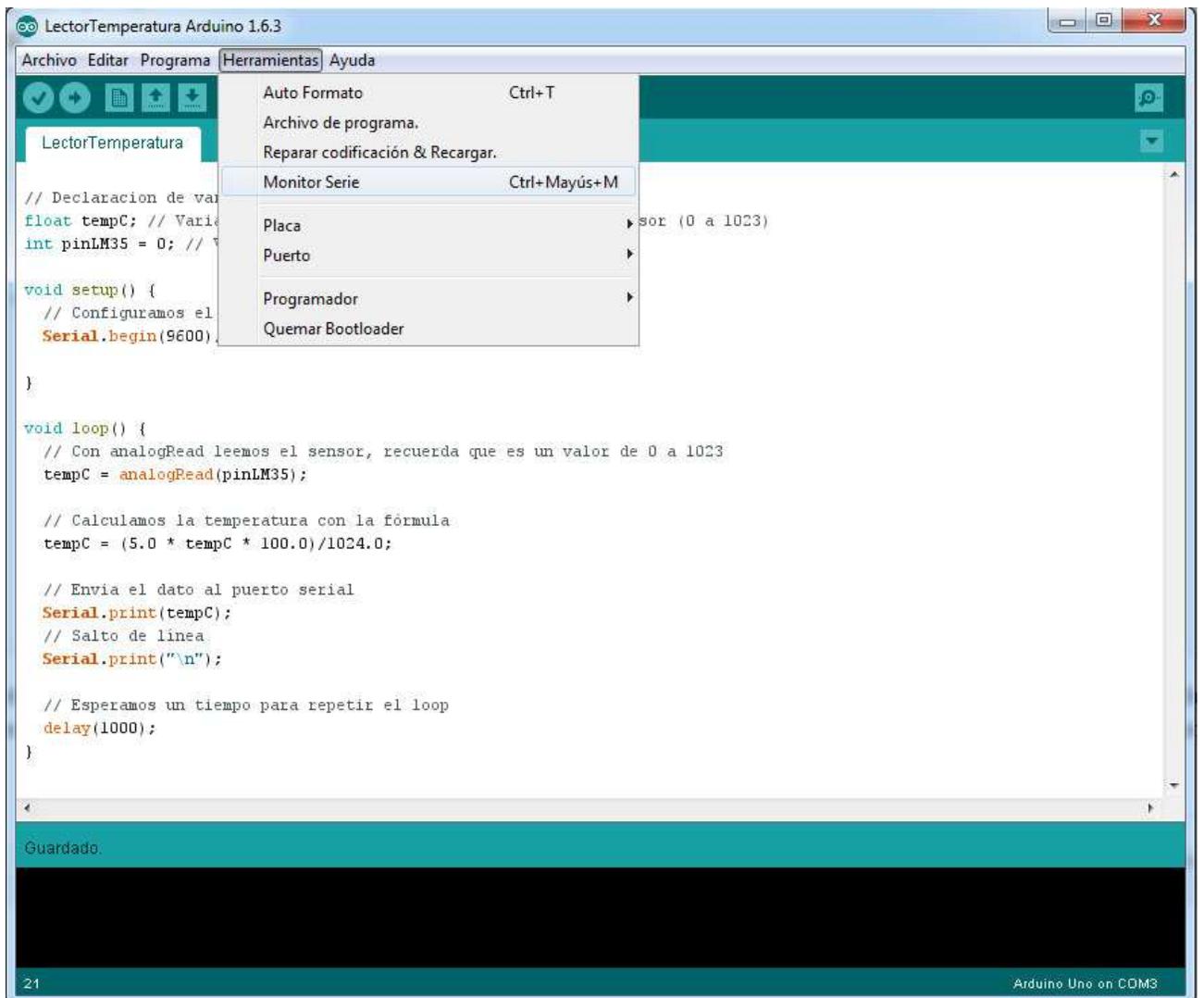
## Conexión LM35

El código que debemos de subir a nuestra placa sería el siguiente.

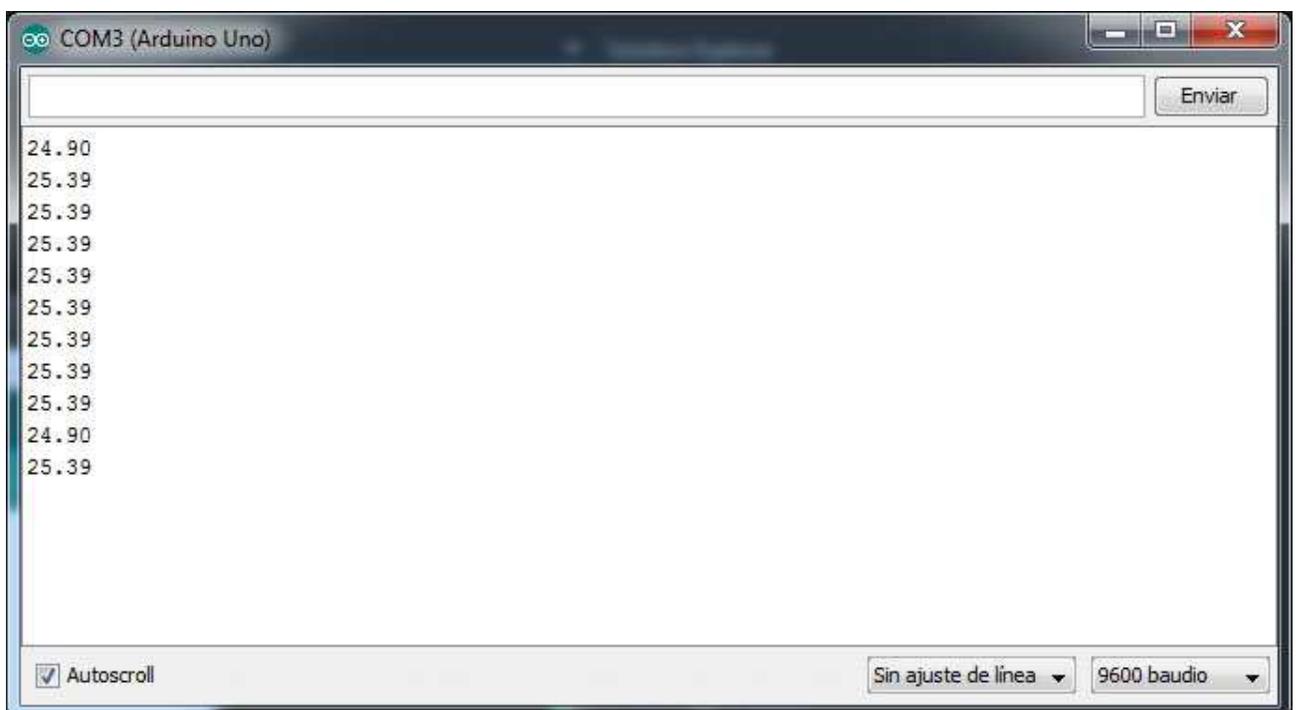
```
/*  
  
Creado: Luis del Valle (ldelvalleh@programarfacil.com)  
  
https://programarfacil.com  
  
*/  
  
// Declaracion de variables globales  
  
float tempC; // Variable para almacenar el valor obtenido del sensor (0 a 1023)  
  
int pinLM35 = 0; // Variable del pin de entrada del sensor (A0)  
  
void setup() {  
  
    // Configuramos el puerto serial a 9600 bps  
  
    Serial.begin(9600);  
  
}  
  
void loop() {  
  
    // Con analogRead leemos el sensor, recuerda que es un valor de 0 a 1023  
  
    tempC = analogRead(pinLM35);  
  
  
    // Calculamos la temperatura con la fórmula  
  
    tempC = (5.0 * tempC * 100.0)/1024.0;  
  
  
    // Envía el dato al puerto serial  
  
    Serial.print(tempC);  
  
    // Salto de línea  
  
    Serial.print("\n");  
  
    // Esperamos un tiempo para repetir el loop  
  
    delay(1000);  
  
}
```

Dos cosas a tener en cuenta son la utilización del **monitor serie y de la función delay**. El puerto serie nos permite ver el resultado de una forma rápida a través del entorno de desarrollo de Arduino. Se utiliza la función begin para iniciar la comunicación con el **monitor serie** y para escribir cada línea utilizamos la función print.

Una vez cargado el código en nuestra placa, puedes ver el monitor serie si accedes a *Herramientas/Monitorserie* o con el atajo de teclado *Ctrl+Mayús+M*. Ver la imagen de donde se encuentra esta opción y como es el monitor serie de Arduino.



Activar monitor serie



Monitor serie

**IMPORTANTE:** Si se desea que se encolumne es mejor usar **Serial.println( )**

**Ejemplo: Serial.println(tempC);**

Por otro lado utilizamos la función *delay* para esperar 1 segundo a volver a tomar la temperatura. Si no esperáramos este tiempo se producirían muchas lecturas en 1 segundo y para este ejemplo no es necesario.

### Mejorando el programa y el funcionamiento

En esta segunda parte del fragmento de código vamos a **cambiar la resolución**. Si utilizas la fórmula que hemos detallado antes te darás cuenta que podemos medir temperaturas de 500°C con 5V. Esto nunca se va a producir ya que se encuentra **fuera del rango** de temperaturas del sensor. Para ello vamos a utilizar la función *analogReference* que nos permite establecer el valor de referencia para la entrada 1023. Hasta ahora hemos trabajado con 5V pero **si utilizamos INTERNAL podremos tener más resolución, 1.1V**. Esto equivale a poder medir temperaturas de hasta 110°C, dentro del rango de operación del LM35. Para ello solo debemos utilizar la siguiente fórmula:

$$\text{Temperatura} = \text{Valor} * 1.1 * 100 / 1024$$

El código que debemos cargar es el siguiente.

```
/*
Creado: Luis del Valle (ldelvalleh@programarfacil.com)
https://programarfacil.com
*/

// Declaracion de variables globales

float tempC; // Variable para almacenar el valor obtenido del sensor (0 a 1023)

int pinLM35 = 0; // Variable del pin de entrada del sensor (A0)

void setup() {

    // Cambiamos referencia de las entradas analógicas

    analogReference(INTERNAL);

    // Configuramos el puerto serial a 9600 bps

    Serial.begin(9600);

}

void loop() {

    // Con analogRead leemos el sensor, recuerda que es un valor de 0 a 1023

    tempC = analogRead(pinLM35);

    // Calculamos la temperatura con la fórmula

    tempC = (1.1 * tempC * 100.0)/1024.0;

    // Envía el dato al puerto serial

    Serial.print(tempC);

    // Salto de línea

    Serial.print("\n");

    // Esperamos un tiempo para repetir el loop

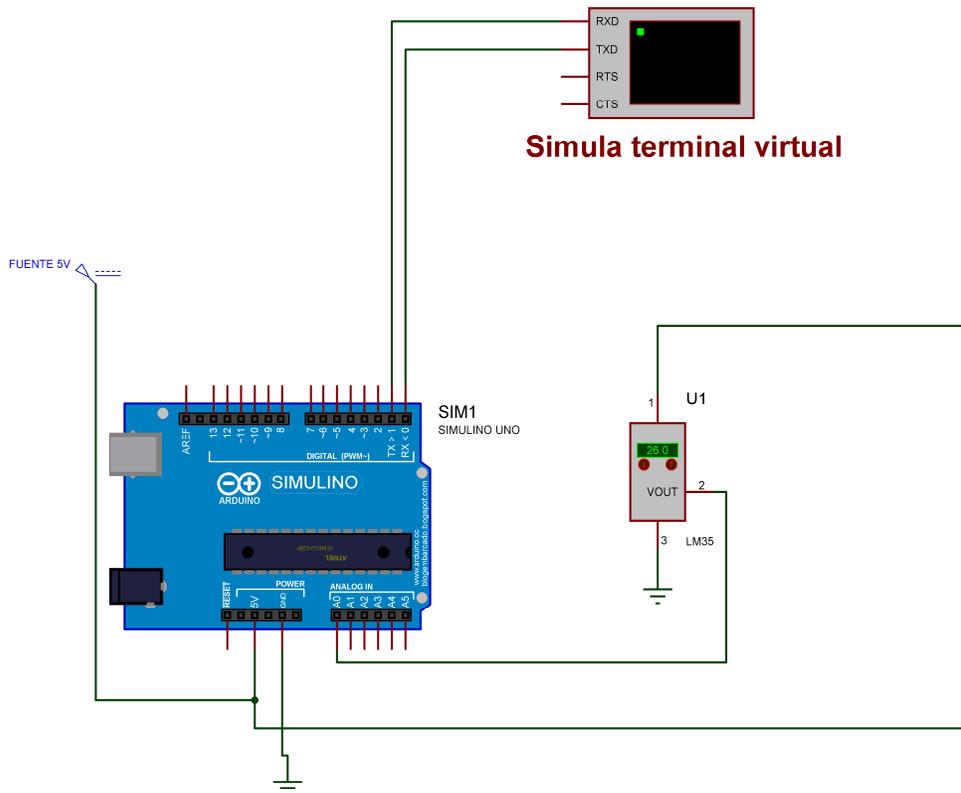
    delay(1000);

}
```

Como puedes comprobar los únicos cambios son la inclusión de la función *analogReference* y el cambio en la fórmula para pasar de 5V a 1.1V.

## Simulación en Proteus

### Sensor de Temperatura LM35 con Arduino - Con salida por terminal virtual Ejemplo1



## Apendice:

### Sensor de temperatura en Arduino

Luis del Valle Hernández

Hoy vamos a hablar de cómo podemos leer la temperatura ambiente. Para ello vamos a utilizar un **sensor de temperatura en Arduino UNO**, más concreto el sensor analógico LM35.

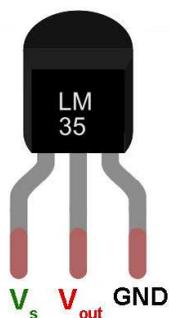
Necesitamos la función `analogRead` para capturar el valor que nos proporciona el sensor.

Recordemos como funcionan las entradas analógicas en Arduino. Aunque estemos hablando de un valor analógico, Arduino discretiza en 1024 valores (0 a 1023). Esto quiere decir que si tenemos un rango de valores de entre 0 V y 5 V, cuando el sensor nos da un valor de 0 V con `analogRead` tendremos un valor de 0 y cuando nos da 5 V tendremos 1023. Si nos da un valor de 2,5 V simplemente tenemos que hacer una regla de 3 y calcular que valor nos proporcionará dicha función.

#### LM35 sensor de temperatura

En este artículo vamos a utilizar el sensor LM35. Este sensor, como cualquier otro sensor, tiene unas características de funcionamiento las cuales se pueden ver en su hoja de datos. Este documento es el punto de partida para utilizar cualquier sensor en Arduino. Nos aportará la información necesaria para trabajar con él y es indispensable consultarlo antes de empezar.

Lo primero que debemos tener claro es el conexaso. Si nos vamos a la página 3 veremos que el sensor tiene 3 patillas, hay que fijarse en el gráfico que pone *LP Package 3-Pin TO-92*. Si miramos de frente el sensor, por la parte plana veremos algo parecido a la siguiente imagen.



Donde **Vs es la alimentación, normalmente a 5 V** que es lo que nos da la placa de Arduino salvo que especifiquemos lo contrario, GND es la toma de tierra y Vout es la salida que proporciona el sensor.

Por lo tanto de aquí podemos sacar como debemos conectar el sensor con Arduino. Antes de hacerlo tenemos que estar seguro que el rango de trabajo de este sensor soporta los 5V que nos va a proporcionar nuestra placa. En caso contrario deberíamos hacer algo para reducir esos 5V. En la página número 1 encontramos un apartado que pone *Features* (características). De aquí podemos sacar mucha información. Si miramos en el apartado que pone *Operates from 4 V to 30 V*, nos está diciendo que funciona con voltajes de **4 V a 30 V** así que estamos dentro del rango permitido.

Ahora vamos a fijarnos en la salida que nos proporciona el LM35. Las dos primeras líneas dicen:

- *Calibrated Directly in Celsius (Centigrade)*
- *Linear + 10-mV/°C Scale Factor*

La primera línea nos dice que el sensor está calibrado en grados centígrados con lo cual no hará falta realizar ninguna transformación si queremos saber la temperatura en estas unidades. La segunda línea nos viene a decir que los cambios de temperatura son lineales con un factor de escala de 10mV por °C (grado centígrado) es decir, para 1 °C tendremos una tensión de 10 mV. **Esto es muy importante ya que hay una regla de proporción entre el voltaje de salida y los grados.**

Si nos fijamos un poco más abajo en las *Features* (características) encontramos el rango completo de temperaturas de -55 °C a 150 °C. Esto quiere decir que a -55 °C tendremos -550 mV y a 150 °C tendremos 1.500 mV. Esto también lo puedes ver en la ficha técnica en la primera página el gráfico que pone *Full-range Centigrade Temperature Sensor*.

Por último hay fijarse también en las *Features*, donde nos dice *0.5°C Ensured Accuracy (at 25°C)* que quiere decir que para garantizar la precisión hay que trabajar en este rango de temperaturas de 0.5°C a 25°C.

Así que ya has podido ver lo importante que son las fichas técnicas de los sensores y dispositivos. Hay que remitirse siempre a la web del fabricante y obtener dicha ficha para sacar el máximo partido a los sensores.

### Calcular la temperatura

Llega la hora de los cálculos.

Hay una función que se llama *analogRead* donde le pasamos como parámetro el número de pin que queremos leer y nos da un valor entre 0 y 1023 o sea, 1024 valores. Esto

quiere decir que si en el pin tenemos 0V nos dará 0 y si tenemos 5V o 3.3V (depende de a que voltaje esté funcionando nuestro Arduino) nos dará 1023, así de sencillo.

Por lo tanto si queremos saber que voltaje tenemos en la entrada analógica solo debemos de multiplicar por 5/1024 (0.0048V de precisión). Esto nos da el voltaje de nuestro sensor, ahora hay que transformar los voltios en grados. Ya hemos averiguado, gracias a la ficha técnica, que 1°C equivale a 10mV (0.01V) por lo tanto solo debemos de dividir el valor de voltaje obtenido antes entre 0.01 y nos dará la temperatura. Para que os quede más claro os dejo aquí la formula que deberíamos aplicar.

$$\text{Temperatura} = (\text{Valor} * 5 / 1024) / 0.01 = \text{Valor} * 5 * 100 / 1024$$

Y con esta fórmula tendríamos la temperatura que nos está facilitando el **sensor LM35**.

## Aumentar la precisión

Como ya hemos visto en la ficha técnica, si queremos garantizar la precisión debemos trabajar en el rango de valores de 0.5°C y 25°C. Esto no quiere decir que nos tengamos que ceñir a estas temperaturas pero si que nos da una idea de con que precisión deberíamos trabajar. Para estos dos valores de grados corresponden 5mV y 250mV.

### ¿Cómo podemos aumentar la precisión?

La función ***analogReference*** que nos permite establecer el valor de referencia para el valor 1023 en los pines analógicos. Esta función puede tomar tres posibles valores:

- **DEFAULT**  
Toma como referencia el valor interno de la placa 3.3V o 5V.
- **INTERNAL**  
Toma como referencia un voltaje interno de 1.1V.
- **EXTERNAL**  
Toma como referencia lo introducido en el pin AREF.

Si dejamos la primera opción, **DEFAULT**, utilizaremos la fórmula que hemos visto antes pudiendo detectar temperaturas de hasta 500°C. No tiene ningún sentido ya que el rango máximo será 150°C. Así que podemos aumentar la precisión de nuestro sensor utilizando la configuración **INTERNAL**. Esto nos permite medir hasta un rango de 110°C. Al cambiar el valor de la referencia, la fórmula cambia, ahora para obtener la temperatura debemos de utilizar la siguiente:

$$\text{Temperatura} = (\text{Valor} * 1.1 / 1024) / 0.01 = \text{Valor} * 1.1 * 100 / 1024$$

Con toda esta información podemos obtener la temperatura de nuestro sensor con simples operaciones matemáticas.