**IMPLEMETACIÓN DE UN DISPOSITIVO DE MEDICIÓN DE TEMPERATURA**

***Versión 1.0***

**Elaboró: Daniel Felipe Barón Espitia**

# OBJETIVO

Dar a conocer el procedimiento para llevar a cabo el diseño de un dispositivo capaz de medir temperatura usando el circuito integrado LM35 y la placa programable Arduino. Además de esto, dar a conocer el código necesario para poder acoplar el circuito integrado a la placa programable y así garantizar la correcta toma de datos.

# ALCANCE

Dar a conocer el procedimiento necesario para poder diseñar un dispositivo capaz de medir temperaturas en un intervalo de −55°C a 150°C con una exactitud de 0.5 °C.

# ESPECIFICACIONES Y RESTRICCIONES DE DISEÑO

El dispositivo medidor de temperatura utiliza un circuito integrado LM35, una LCD y una placa programable Arduino. Los dispositivos antes mencionados son descritos a continuación:

**CIRCUITO INTEG**RADO

El circuito integrado LM35 es un sensor de temperatura con una salida de voltaje linealmente proporcional a la temperatura centígrada. Esto quiere decir que, el aumento lineal en la salida de voltaje del sensor equivale a un aumento lineal de la temperatura medida con el sensor.

En la figura 1 se puede observar la configuración de los pines del circuito integrado LM35, sin embargo, es importante revisar el datasheet del circuito integrado que se vaya a utilizar para realizar el procedimiento para evitar malas conexiones y por ende un mal funcionamiento del dispositivo.

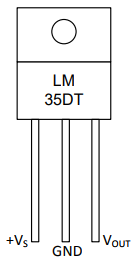
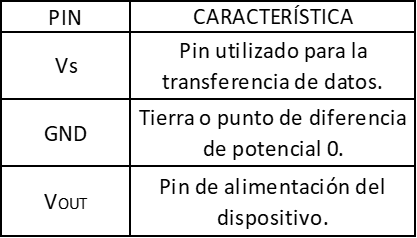


Figura 1: Configuración de los pines del circuito integrado LM35.

Tabla 1: Descripción de los pines del circuito integrado LM35.



Los datos proporcionados por el LM35 no son datos de temperatura sino de voltaje, por lo tanto, es necesario realizar una conversión para que la información presentada tenga las unidades de temperatura. Para esto es necesario utilizar la información proporcionada por los fabricantes, la cual se encuentra en el datasheet de nuestro dispositivo.

Por último, es importante aclarar que para tomar valores positivos es necesario realizar otra topología en el circuito, sin embargo, en esta ocasión solo se hará énfasis en la toma de datos de temperatura positivos.

## PLACA PROGRAMABLE

La placa programable utilizada en el diseño del dispositivo es un Arduino Mega 2560, sin embargo, es posible utilizar cualquier otra versión de este hardware debido a que para la implementación del sensor únicamente es necesario los pines digitales, los pines análogos, los pines de alimentación y los pines de GND, y todas las placas cuentan con estos tipos de pines. Las variaciones entre los diferentes modelos de Arduino son la cantidad de pines, el tamaño y las librerías que pueden ser utilizadas, pero todos cuentan con una línea básica en cuanto a la funcionalidad. La principal característica del Arduino Mega 2560 es su gran cantidad de pines (53 pines digitales, 15 pines análogos, entre otros).

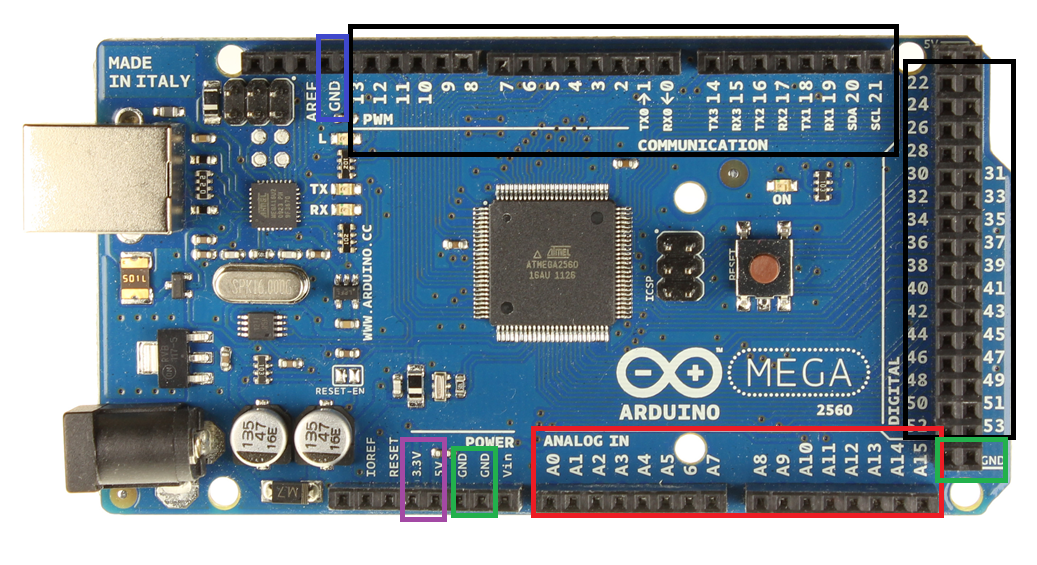


Figura 2: Esquematización de los diferentes pines que posee un Arduino Mega 2560.

**Pines digitales (Recuadro Negro):** Una señal digital es aquella que solo puede tomar dos valores, 1 o 0 en el caso de los pines de Arduino. Este tipo de pines son utilizados cuando el estado de un dispositivo puede tomar únicamente dos valores, encendido o apagado. Usualmente estos pines son utilizados para enviar información, como por ejemplo cambiar el estado de un LED de encendido a apagado o viceversa.

**Pines Análogos (Recuadro Rojo):** Una señal eléctrica analógica es aquella en la que los valores de la tensión o voltaje varían constantemente y pueden tomar cualquier valor. Por ende, estos pines son útiles cuando la toma de datos o la información externa que recibe toma valores diferentes, como por ejemplo el sensor de temperatura que se va a utilizar. Usualmente se conectan a estos pines dispositivos que envían información, y por ende el Arduino la recibe y la procesa.

**Pines de Alimentación (Recuadro Morado):** Estos pines son utilizados para alimentar los diferentes dispositivos externos, por ejemplo, pantallas LCD, LED’s, entre otros. El Arduino Mega 2560 cuenta con dos pines de alimentación, una que maneja un voltaje de 3.3V y otra que maneja un voltaje de 5V. Es importante revisar el datasheet del fabricante si se desea conectar algún dispositivo externo, esto con el fin de evitar daños o mal funcionamiento de los correspondientes dispositivos.

**Pines de GND:** Pin a tierra o punto de voltaje cero.

## LCD (Liquid Crystal Displays)

Las pantallas de cristal líquido (en inglés “Liquid Crystal Displays” o LCD’s) ofrecen una manera muy rápida y vistosa de mostrar información en forma de texto. Las LCD’s de caracteres más habituales son de 4-bit o 8-bit, dependiendo del número de cables (bits) que necesitan tener conectados al circuito para poder recibir o enviar datos.

Cada modelo de LCD es diferente, por lo que es muy importante consultar su datasheet concreto para poder distinguir los diferentes pines de conexión que ofrece y sus características generales. Sin embargo, lo más habitual es que una LCD estándar ofrezca un pin para recibir la alimentación y otro pin para conectar la pantalla a tierra, un pin para regular el contraste de la pantalla, tres pines de control generalmente marcados como “RS”, “EN” y “RW”, varios pines para establecer las líneas de comunicación en paralelo y, por último, dos pines exclusivos para el circuito de la luz de fondo. (Artero, O. T., 2013)

****

Figura 3: Representación de los pines de una LCD de 16x2 de 4 bits.

Tabla 2: Descripción de los pines de una LCD de 16x2 de 4 bits.

|  |  |
| --- | --- |
| **PIN** | **CARACTERÍSTICA** |
| GND | Tierra o punto de diferencia de potencial 0. |
| VCC | Pin de alimentación del dispositivo. |
| Vo | Pin para configurar el contraste de la LCD. |
| RS | Pin que sirve para que el microcontrolador le diga a la LCD si quiere mostrar caracteres o si lo que quiere es enviar comandos de control. |
| R/W | Pin para determinar si la pantalla va a proporcionar o a recibir información. |
| E | Pin para advertir a la LCD que el microcontrolador le va a enviar datos. |
| D0-D7 | Se usan para establecer las líneas de comunicación en paralelo por donde se transfieren los datos y los comandos de control de la placa Arduino hacia el LCD. |
| A | Pin utilizado para controlar la luz de fondo, este pin es conectado a Voltaje, usualmente 5 V. |
| K | Pin utilizado para controlar la luz de fondo, este pin es conectado a GND. |

# PASO A PASO PARA LA CONEXIÓN DE LOS DISPOSITIVOS

Para construir el diseño necesitaremos los siguientes materiales:

* Protoboard
* Placa Arduino Mega 2560
* Circuito integrado LM35
* LCD de 16x2 de 4 bits
* Potenciómetro de 5K

1. El primer paso a realizar es conectar los pines de alimentación del Arduino a la Protoboard. Es importante usar una convención específica para la conexión a los pines de voltaje y GND. Usualmente se usa un cable negro para representar GND y un cable rojo para representar Voltaje.

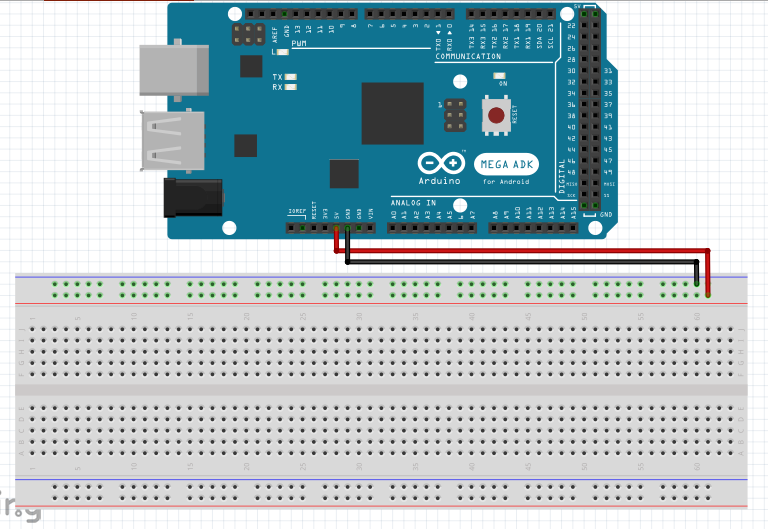


Figura 4: Conexión de voltaje y GND desde el Arduino hasta la Protoboard.

1. Seguido de esto, se realiza la conexión del circuito integrado LM35 al Arduino. Se conecta el pin Vs a un analógico del Arduino. En este caso fue el pin A1. Es importante conectar el pin Vs a un pin análogo debido a que la información que manda el LM35 varia constantemente y puede tomar un valor en un rango de −55°C a 150°C. El pin Vout se conecta al camino de voltaje de la Protoboard y, por último, se conecta el pin de GND al camino de GND de la Protoboard.

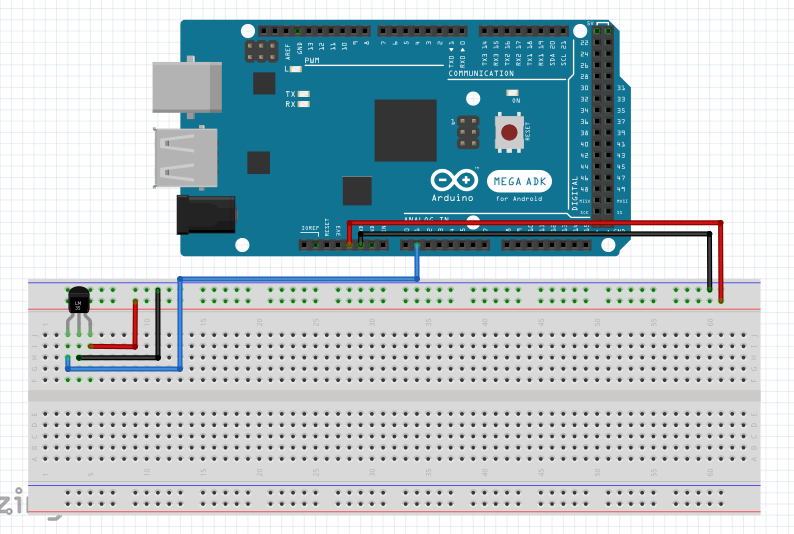


Figura 5: Conexión de los pines del LM35 (cable azul es para el pin de transferencia de datos).

1. Para poder representar la información de forma gráfica se usa una LCD de 16x2. Primero se conectan los pines de GND y VCC a los pines de tierra y voltaje de la Protoboard respectivamente. Este procedimiento se observa en la Figura 5.

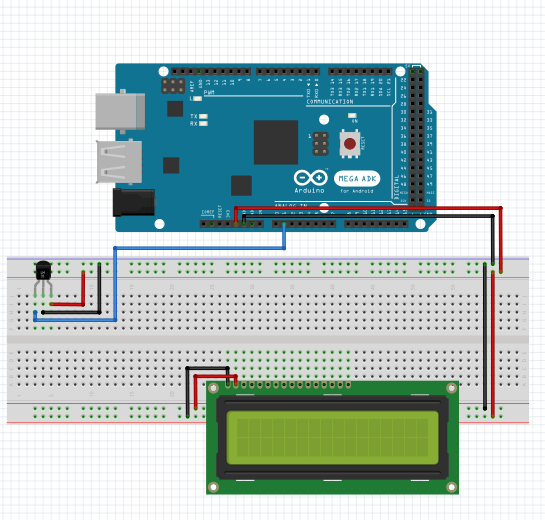


Figura 6:Conexión de los pines GND y VCC de la LCD.

1. La LCD que se usa en este procedimiento tiene la ventaja de permitir la variación del contraste por medio de la utilización de un potenciómetro, el cual consiste en una resistencia variable, la cual se ajusta con una perilla. El potenciómetro es conectado al pin Vo de la LCD. A su vez, el potenciómetro debe estar conectado a tierra y a voltaje de la Protoboard. El cable de color Cian representa el pin por el cual el potenciómetro envía la señal de resistencia variable. En la figura 7 se representa el esquemático de los pines para un potenciómetro convencional.

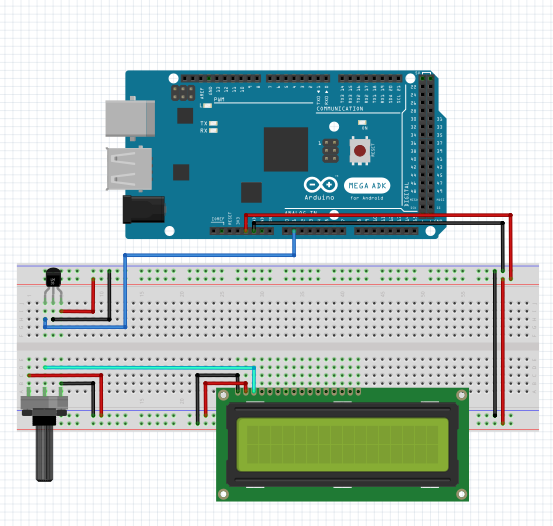


Figura 7: Conexión del pin de regulación de contraste de la LCD.



Figura 8: Esquemático de los pines de un potenciómetro.

1. Luego de conectar el potenciómetro al pin Vo de la LCD, se conectan los pines correspondientes al ánodo y el cátodo para alimentar la luz de fondo de la LCD. Estos dos pines se conectan a 5V y a GND, respectivamente.

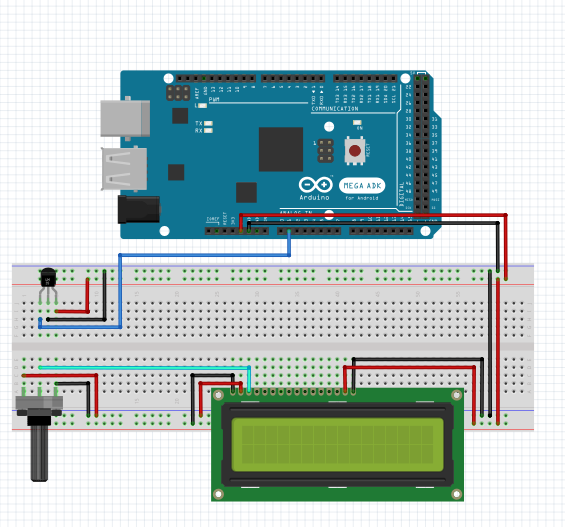


Figura 9: Conexión para la iluminación del fondo de la LCD (A y K).

1. Como anteriormente se mencionó, los pines RS, RW y E son pines utilizados para de algún modo avisarle a la LCD que va a recibir información, por lo cual es necesario conectarlos a entradas digitales del Arduino. Sin embargo, el pin E (enable) es conectado directamente a tierra.

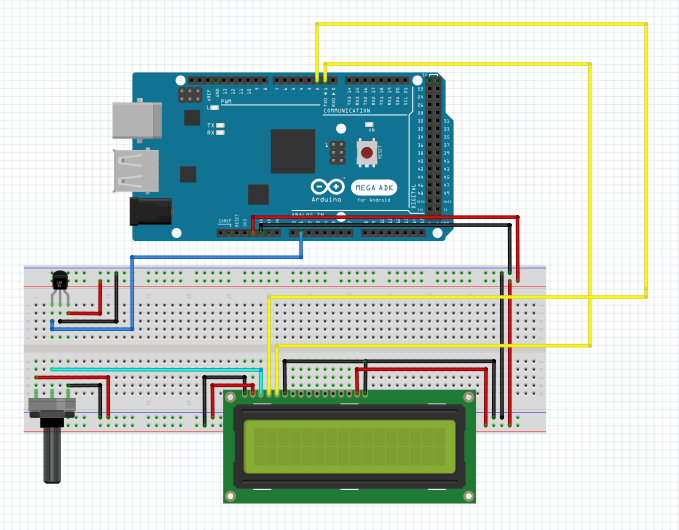


Figura 10: Conexión de los pines RS, RW y E de la LCD.

1. Como anteriormente se mencionó, los pines D0 a D7 son usados para la transmisión de la información y los comandos con los cuales se controlan las funciones de la LCD. Por otra parte, únicamente es necesario utilizar los pines de D4 a D7 para transmitir información. Para transmitir este tipo de información se utilizan los pines digitales de Arduino.

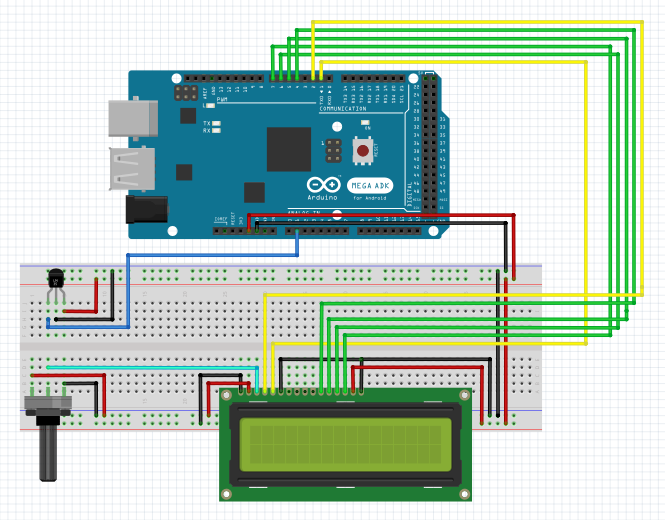


Figura 11: Conexión de los pines D4 a D7 de la LCD de 16x2.

## Construcción del código

Hasta este instante hemos construido la parte del Hardware de nuestro dispositivo de medición de temperatura, de aquí en adelante nos enfocaremos en la elaboración del código para poder leer, procesar y presentar la información recolectada con el circuito integrado LM35. Para realizar el código, necesitaremos del software de Arduino. Luego de haber instalado el software, se procede a abrirlo, una vez realizado esto aparecerá la siguiente ventana.

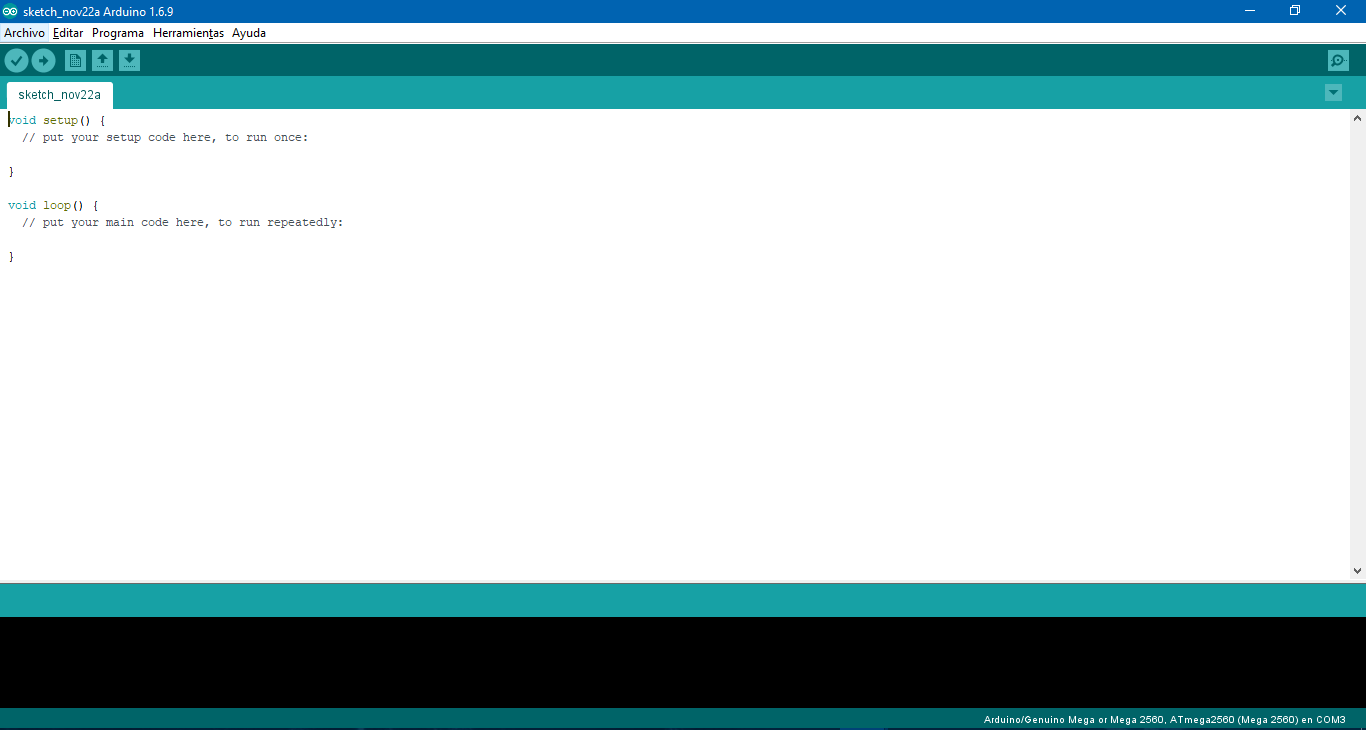


Figura 12: Interfaz de programación de Arduino.

Como se puede observar, tenemos dos bloques de código, el primero es el *void setup(),* en el cual se escribirán todas declaraciones de variables y constantes debido a que este bloque solo se ejecuta una vez. El segundo bloque observado es el *void loop(),* en el cual se escribirán todas las funciones y en general el código que se va a ejecutar cíclicamente.

Para comenzar con el código, es necesario cargar las librerías correspondientes para poder utilizar la LCD. La correspondiente librería se llama LiquidCrystal. Seguido de esto, se declara una variable de tipo “LiquidCrystal” de la siguiente manera:

*LiquidCrystal nombreLCD (rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)*



Figura 13: Código utilizado para acoplar el Arduino, la LCD y el LM35.

Para poder imprimir en la LCD los datos reales de temperatura es necesario hacer una conversión de voltios a grados centígrados. Esta conversión se hace por medio de la siguiente relación:

(1)

Esta conversión es debido a el tipo de información que proporciona el LM35, la cual equivale a un numero de 0 a 1024, y es dependiente del voltaje. Como nuestro Arduino maneja un voltaje de 5 V, por regla de 3 se calcula el valor del voltaje entregado por el LM35. Por otra parte, del datasheet del circuito integrado se sabe que la pendiente o valor de conversión de voltios a grados centígrados es que por el aumento de 1°C el LM35 aumentará 10 mV. Además, se sabe que 10 mV son equivalente a 0.01 V, por lo tanto, la función de conversión de voltaje a grados centígrados es equivalente a la ecuación (1).

# REFERENCIAS

Artero, O. T. (2013). Arduino: curso práctico de formación. México D.F.: Alfaomega.

# ENLACES

## Librería para la LCD 16x2 LiquidCrystal:

Versiones de Arduino anteriores a 1.0: <https://playground.arduino.cc/uploads/Main/LiquidCrystal.zip>

Versiones de Arduino posteriores a 1.0: <https://playground.arduino.cc/uploads/Main/LiquidCrystal_1.zip>

Datasheet LM35:

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

Descargar Arduino ultima versión:

<https://www.arduino.cc/en/Main/Software>

# CONTROL DE CAMBIOS

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN DEL CAMBIO** | **FECHA** | **VERSIÓN** | **APROBADO POR** |
|  |  |  |  |