

TP_7

VISUALIZACIÓN DEL CONTROL DE TEMPERARATURA
ATRAVES DE UNA PANTALL LCD

AUTOR: AHMED CHATOU BOUGDADI

1. INTRODUCCIÓN

El objetivo de esta práctica es modelar y programar un termómetro para exteriores, con el cual mediremos la temperatura ambiente en cada momento. Para ello representaremos los datos medidos mediante un sensor de temperatura KY-001 en un LCD de 16x2.

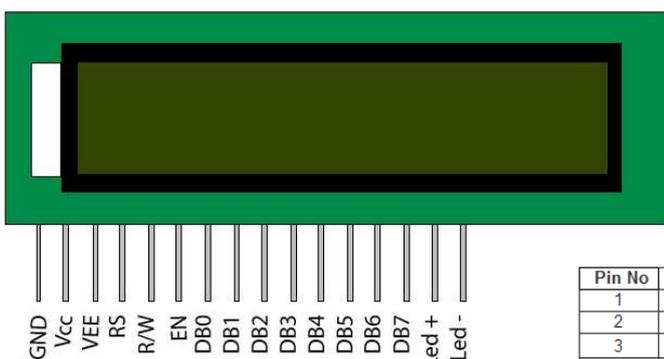
2. MATERIALES Y COMPONENTES

- Protoboard
- Placa Arduino UNO
- Sensor de temperatura digital KY-001
- Potenciómetro (resistencia Variable)
- Resistencia de 220 ohmios
- LCD de 16 x 2
- Jumpers de conexión (Cables)

3. DATASHEET LCD DISPLAY

La pantalla LCD (pantalla de cristal líquido) es un módulo de pantalla electrónica y encuentra una amplia gama de aplicaciones. Una pantalla LCD de 16x2 es un módulo muy básico y se usa muy comúnmente en varios dispositivos y circuitos. Estos módulos son preferidos en siete segmentos y otros LED de segmentos múltiples. Las razones son: las pantallas LCD son económicas; fácilmente programable; no tienen ninguna limitación para mostrar caracteres especiales e incluso personalizados (a diferencia de los siete segmentos), animaciones, etc. Una pantalla LCD de 16x2 significa que puede mostrar 16 caracteres por línea y hay 2 de esas líneas. En esta pantalla LCD, cada personaje se muestra en una matriz de 5x7 píxeles. Esta pantalla LCD tiene dos registros, a saber, Comando y Datos.

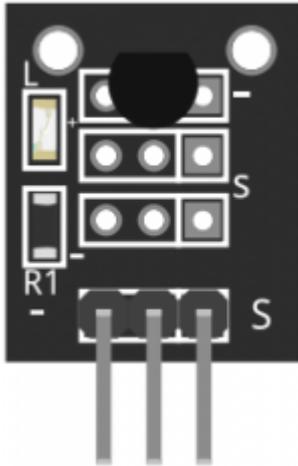
El registro de comando almacena las instrucciones de comando dadas a la pantalla LCD. Un comando es una instrucción dada a la LCD para realizar una tarea predefinida, como inicializarla, borrar su pantalla, establecer la posición del cursor, controlar la visualización, etc. El registro de datos almacena los datos que se mostrarán en la pantalla LCD. Los datos son el valor ASCII del carácter que se mostrará en la pantalla LCD. Haga clic para obtener más información sobre la estructura interna de una pantalla LCD.



Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	VEE
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight Vcc (5V)	Led+
16	Backlight Ground (0V)	Led-

3. DATASHEET SENSOR TEMPERATURA KY-001

El módulo de sensor de temperatura Keyes KY-001 o KY-ds18b20, permite la medición de la temperatura ambiente utilizando el bus serie digital o un display LCD de 16x2.



El módulo de sensor de temperatura KY-001 consta de un sensor de temperatura digital de bus único DS18B20, un LED y una resistencia. Compatible con plataformas de electrónica populares como Arduino, Raspberry Pi y Esp8266.

Especificaciones:

Voltaje de operación	3.0V to 5.5V
Rango medición de temperatura	-55°C to 125°C [-57°F to 257°F]
Rango de precisión de medición	±0.5°C
Dimensiones	18.5mm x 15mm [0.728in x 0.591in]

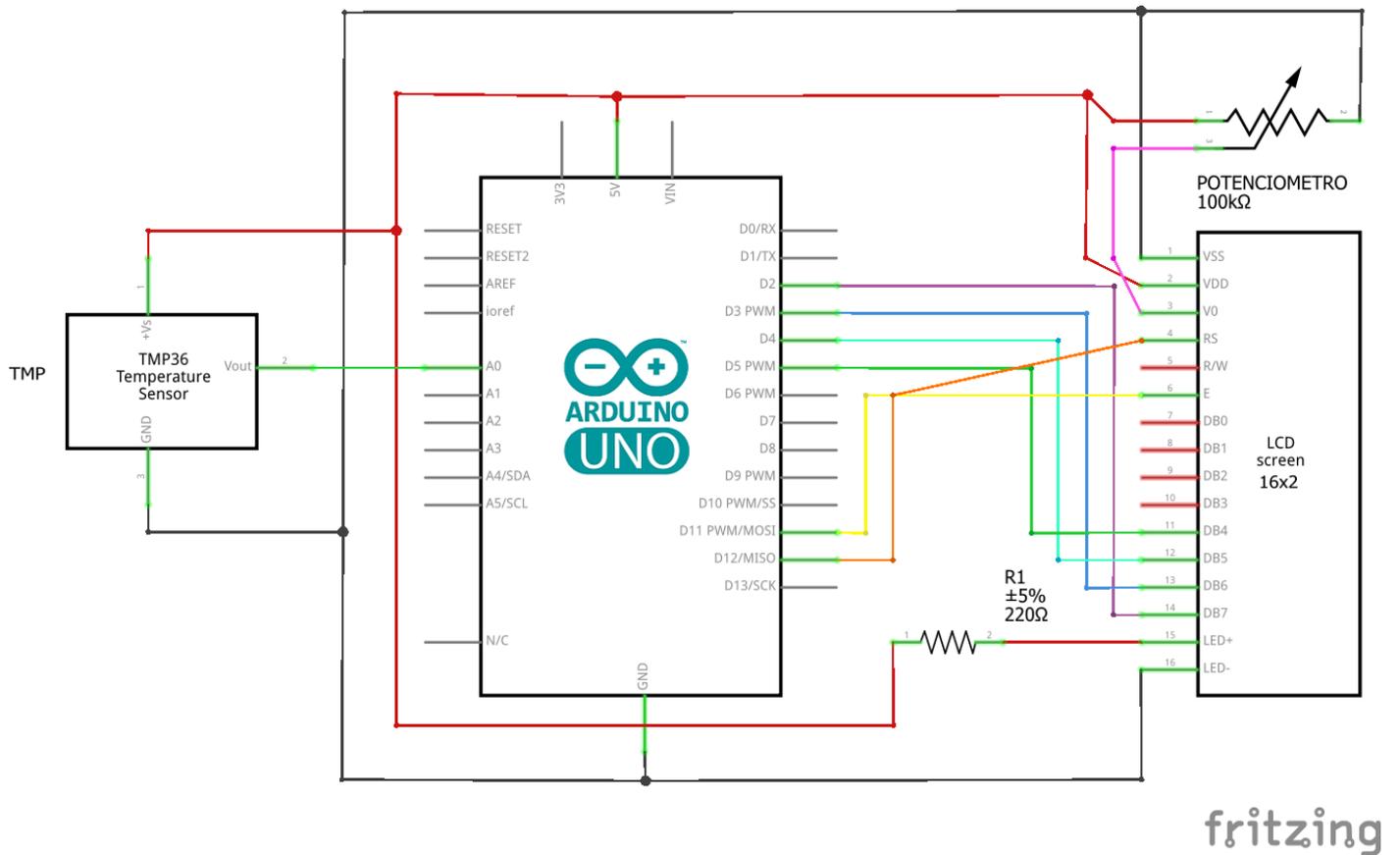
Conexión:

Conecte la línea de alimentación (medio) y tierra (-) a +5 y GND, respectivamente. Conecte la señal (S) al pin 2 en el Arduino.

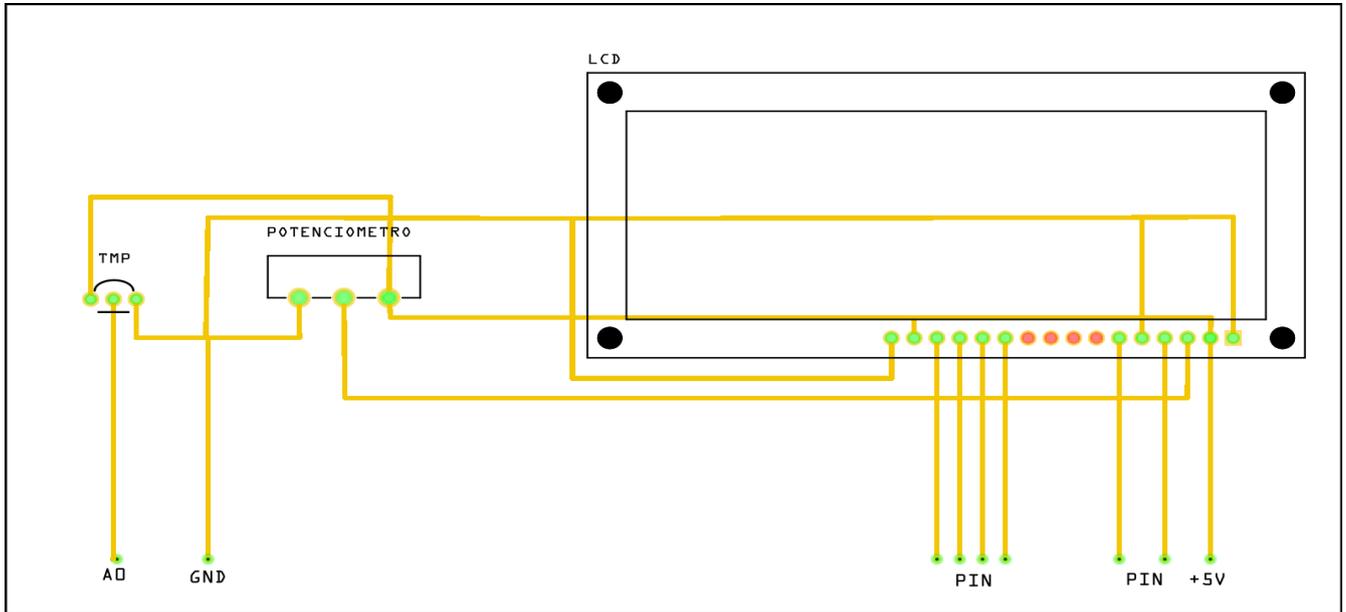
KY-001	Arduino
S	Pin 2
medio	+5V
-	GND

4. ESQUEMATICO

Esquemático de la conexión del circuito.

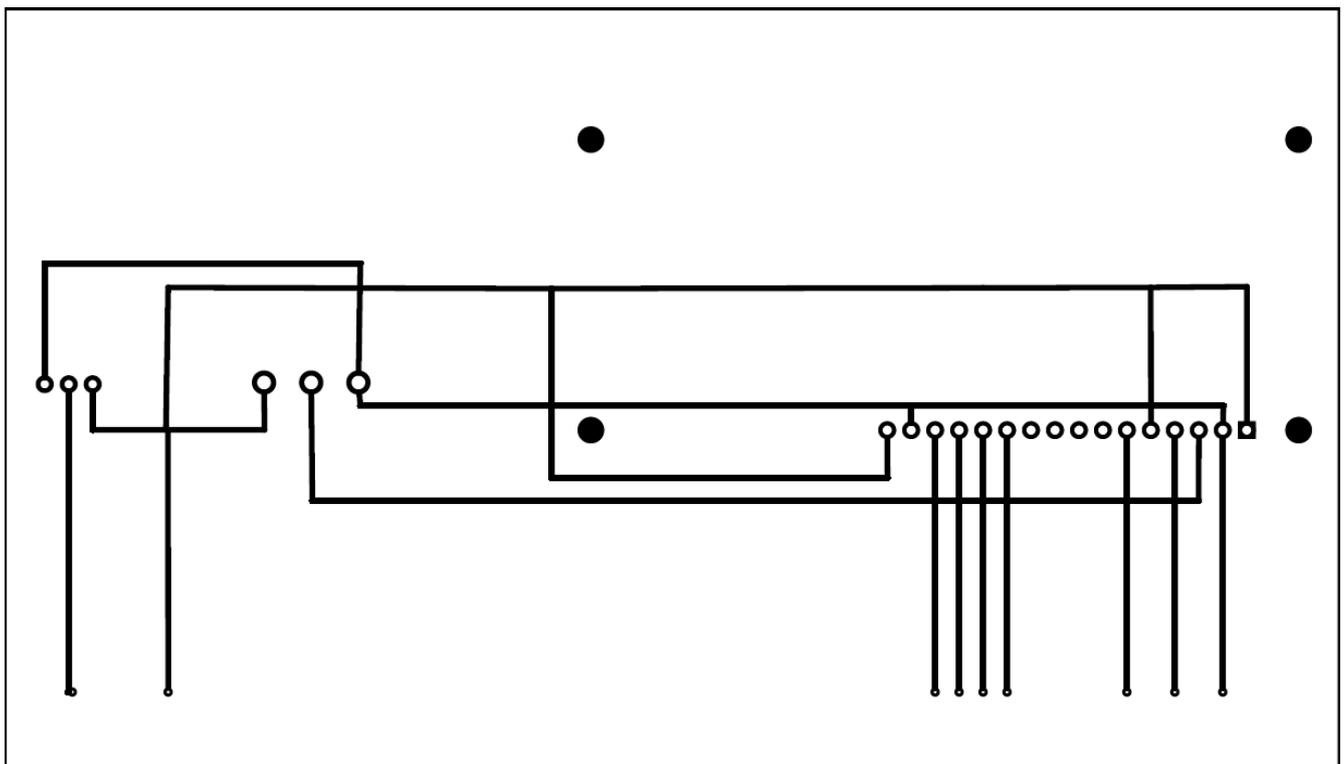


5. PLACA PCB DEL CIRCUITO Y CÓDIGO G

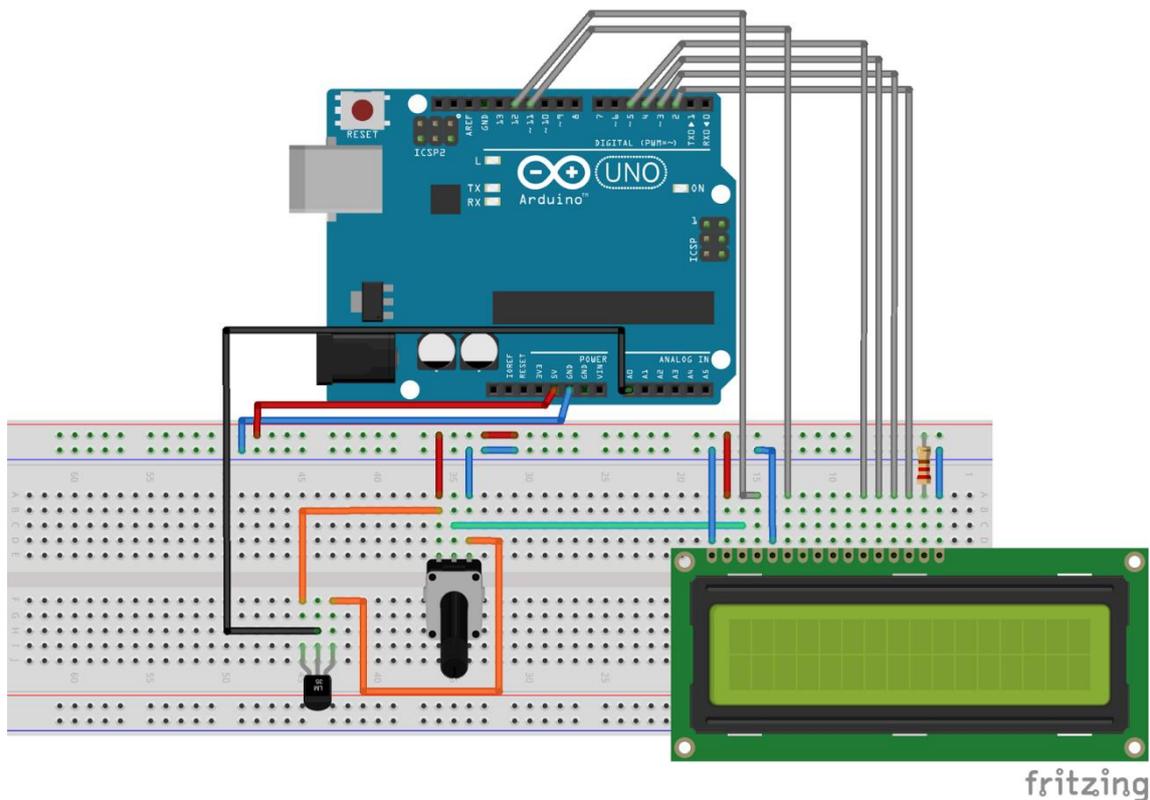


fritzing

6. CIRCUITO MONTADO EN FRITZING



fritzing



Para comenzar nuestro montaje prepararemos la protoboard. Primeramente, conectaremos un cable uniendo nuestro polo positivo de la protoboard con el PIN 5V. El negativo de la protoboard lo conectaremos con GND en nuestro controlador. Por último y para evitar problemas futuros realizaremos dos puentes en la mitad de la board para unir las filas de positivos entre sí y de negativos entre sí.

Ahora que ya tenemos la protoboard preparada colocaremos la LCD. Para ello la LCD debe prepararse previamente para poder conectarla a una protoboard. Si te perdiste éste tutorial puedes acceder a él [aquí](#).

Y por último, nos quedará la conexión del LCD a nuestro controlador. Los pines de la LCD 1, 2, 5, 15 y 16 van conectados a la protoboard (empezando por la izquierda). El 1, 5 y 16 van conectado al negativo de la protoboard (GND). El 2 va conectado mediante un cable a la fila positiva de la protoboard (5V). Por último, el 15 va conectado mediante la resistencia de 220 ohmios a la fila positiva (5V). Los pines 4, 6, 11, 12, 13, 14 irán conectados a nuestro Arduino de la siguiente forma:

4 LCD -> PIN 12

6 LCD -> PIN 11 (PWM)

11 LCD -> PIN 5 (PWM)

12 LCD -> PIN 4

13 LCD -> PIN 3 (PWM)

14 LCD -> PIN 2.

7. CÓDIGO

El código ha sido adaptado para las pantallas que no cuentan con una interface de conexión rápida.

```
#include <LiquidCrystal.h>

const int TMP36Pin = A0; //Conectamos pin central a A0
float temperatura = 0; //Almacenamos la medida de la temperatura

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //LCD display sin interface de
conexion

void setup() {
  Serial.begin(9600); //configuramos la conexion con el PC
  lcd.begin(16, 2); //inicia el Display LCD
  lcd.print("hello, world!"); //Mensaje de inicio al encender el
circuito y la pantalla
  delay (1000);
}

void loop() {
  int lectura = analogRead(TMP36Pin); //leer el sensor
  float voltaje = lectura *5.0 / 1024.0; //Voltaje entre 0 y 5V
  temperatura = (voltaje - 0.5) * 100; //Dato de temperatura en °
Celsius

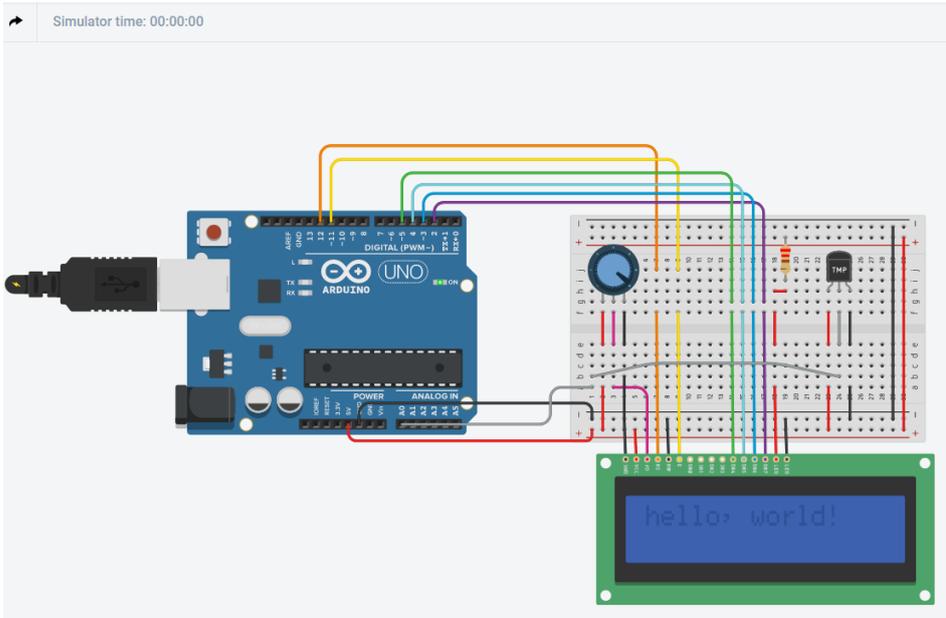
  lcd.clear(); //Borramos la pantalla para
luego actualizar datos
  lcd.setCursor(0,0); //Ponemos cursor en 0
  lcd.print("Temperatura:"); //Mostramos los datos
obtenidos por el sensor
  lcd.print(temperatura);
  lcd.print("C"); //Algunos de los caracteres
dan error (como el °)

  Serial.print ("Temperatura:");
  Serial.print(temperatura); //Mostramos la tempoeratura
por medio de Serial Monitor
  Serial.println("C");

  delay(1000); //Esperamos para realizar la
siguiente medida
}
```

8. SIMULACIÓN EN TINKERCAD

Medición de la temperatura con los parámetros que ofrece el simulador. <https://www.tinkercad.com/things/a5ArJCfIkLU>

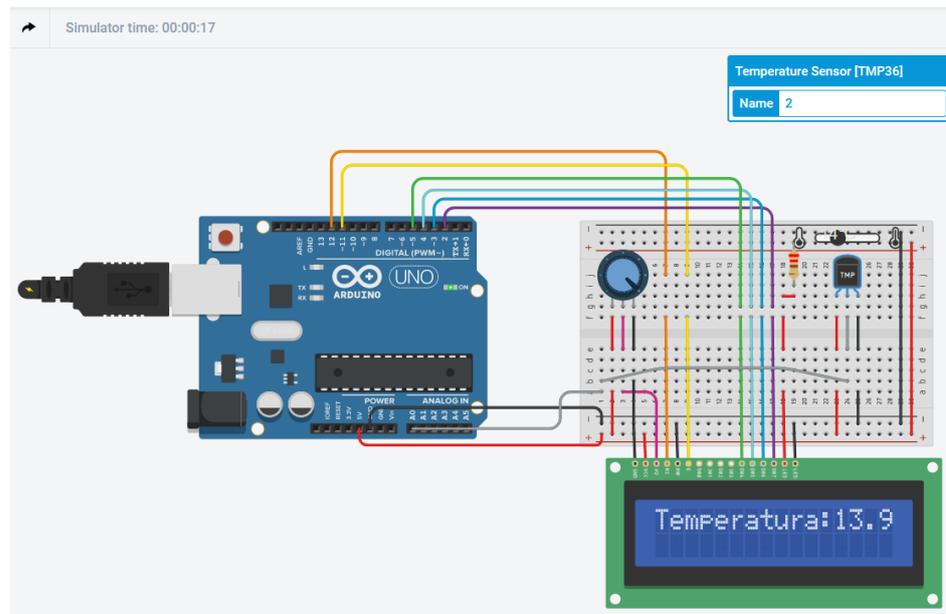


Simulator time: 00:00:00

```
Code Stop Simulation Export Share
```

Text

```
1 #include <LiquidCrystal.h>
2
3 const int TMP36Pin = A0; //Conectamos pin central a A0
4 float temperatura = 0; //Almacenamos la medida de la temperatura
5
6 LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //LCD display sin interface de
7
8 void setup() {
9   Serial.begin(9600); //configuramos la conexion con el PC
10  lcd.begin(16, 2); //inicia el Display LCD
11  lcd.print("hello, world!"); //Mensaje de inicio al encender el circ
12  delay(1000);
13 }
14
15 void loop() {
16   int lectura = analogRead(TMP36Pin); //leer el sensor
17   float voltaje = lectura * 5.0 / 1024.0; //Voltaje entre 0 y 5V
18   temperatura = (voltaje - 0.5) * 100; //Dato de temperatura en ° Ce
19
20
21   lcd.clear(); //Borramos la pantalla para lu
22   lcd.setCursor(0,0); //Fonemos cursor en 0
23   lcd.print("Temperatura:"); //Mostramos los datos obtenid
24   lcd.print(temperatura);
25   lcd.print("C"); //Algunos de los caracteres da
26
27   Serial.print ("Temperatura:");
28   Serial.print(temperatura); //Mostramos la temperatura po
29   Serial.println("C");
30
31   delay(1000); //Esperamos para realizar la si
32 }
33
34
35
```



Simulator time: 00:00:17

Temperature Sensor [TMP36]
Name 2

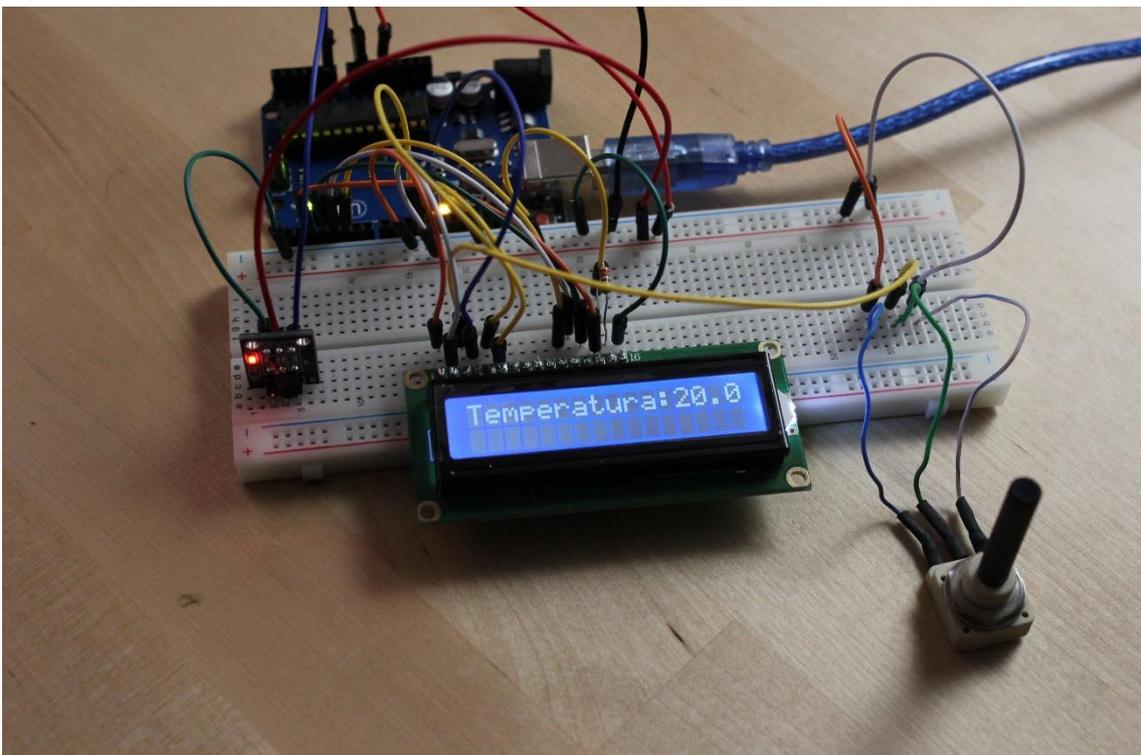
```
Code Stop Simulation Export Share
```

Text

```
1 #include <LiquidCrystal.h>
2
3 const int TMP36Pin = A0; //Conectamos pin central a A0
4 float temperatura = 0; //Almacenamos la medida de la temperatura
5
6 LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //LCD display sin interface de
7
8 void setup() {
9   Serial.begin(9600); //configuramos la conexion con el PC
10  lcd.begin(16, 2); //inicia el Display LCD
11  lcd.print("hello, world!"); //Mensaje de inicio al encender el circ
12  delay(1000);
13 }
14
15 void loop() {
16   int lectura = analogRead(TMP36Pin); //leer el sensor
17   float voltaje = lectura * 5.0 / 1024.0; //Voltaje entre 0 y 5V
18   temperatura = (voltaje - 0.5) * 100; //Dato de temperatura en ° C
19
20
21   lcd.clear(); //Borramos la pantalla para l
22   lcd.setCursor(0,0); //Fonemos cursor en 0
23   lcd.print("Temperatura:"); //Mostramos los datos obtenid
24   lcd.print(temperatura);
25   lcd.print("C"); //Algunos de los caracteres d
26
27   Serial.print ("Temperatura:");
28   Serial.print(temperatura); //Mostramos la temperatura p
29   Serial.println("C");
30
31   delay(1000); //Esperamos para realizar la s
32 }
33
34
35
```

9. ANEXOS

9.1. FOTOS Y VIDEOS DEL MONTAJE



9.2. ENLACES DE WEBS UTILIZADAS

Las webs utilizadas han sido:

tinkercad para la simulación y montaje del circuito y para obtener más información sobre la practica se han visitado .

<https://www.tinkercad.com>

<https://arduinoab.wordpress.com/>