

## PROPUESTA ARDUINO por José Manuel Aranga Salazar

### Descripción del proyecto

#### CÁLCULO DE LAS ENTALPÍAS DE DISOLUCIÓN DE VARIAS SALES

En el presente proyecto vamos a realizar un montaje para el cálculo de las entalpías de disolución de diferentes solutos.

Para ello vamos a emplear los siguientes materiales:

- Placa Arduino UNO
- Protoboard
- Pulsador
- Led rojo y led de la placa
- Resistencias (220  $\Omega$  para el led y 10 k $\Omega$  para el pulsador)
- Sensor de temperatura TMP36
- Cableado
- Tubo de ensayo
- Calorímetro
- Agua destilada
- Distintas sales
- Balanza
- Tapón de goma
- Taladratapones
- Estaño para soldar
- Soldador

Vamos a construir un sistema en el que el sensor de temperatura TMP36 va a darnos la temperatura que hay en el interior de un calorímetro.

La ventaja de usar este sensor está en que nos da una salida de voltaje directamente proporcional a la temperatura en grados Celsius mediante el empleo de la siguiente ecuación:

$$\text{voltaje} = \left( \frac{\text{sensorValor}}{1024.0} \right) * 5.0$$

Y para convertirlo en temperatura (Celsius) empleamos esta otra:

$$\text{temperatura} = (\text{voltaje} - 0.5) * 100$$

Dado que dicho sensor no es sumergible, vamos a hacer un pequeño montaje en que los cables que van del sensor a la protoboard van a ir soldados a cada uno de los tres pines. El sensor irá dentro de un tubo de ensayo de vidrio que lo protegerá, y estará cerrado con un tapón de goma horadado mediante el taladratapones. Por dicho orificio pasarán los tres cables hacia la protoboard. Los cables deberán tener unos 30 – 40 cm para permitir introducir el sensor en el tubo de ensayo y este, a su vez, en el calorímetro.

Con este sistema se podrán calcular no sólo las indicadas entalpías, sino que también se podrían calcular las de distintas reacciones químicas.

Los resultados del cálculo de temperatura y de entalpía saldrán por el Serial Monitor.

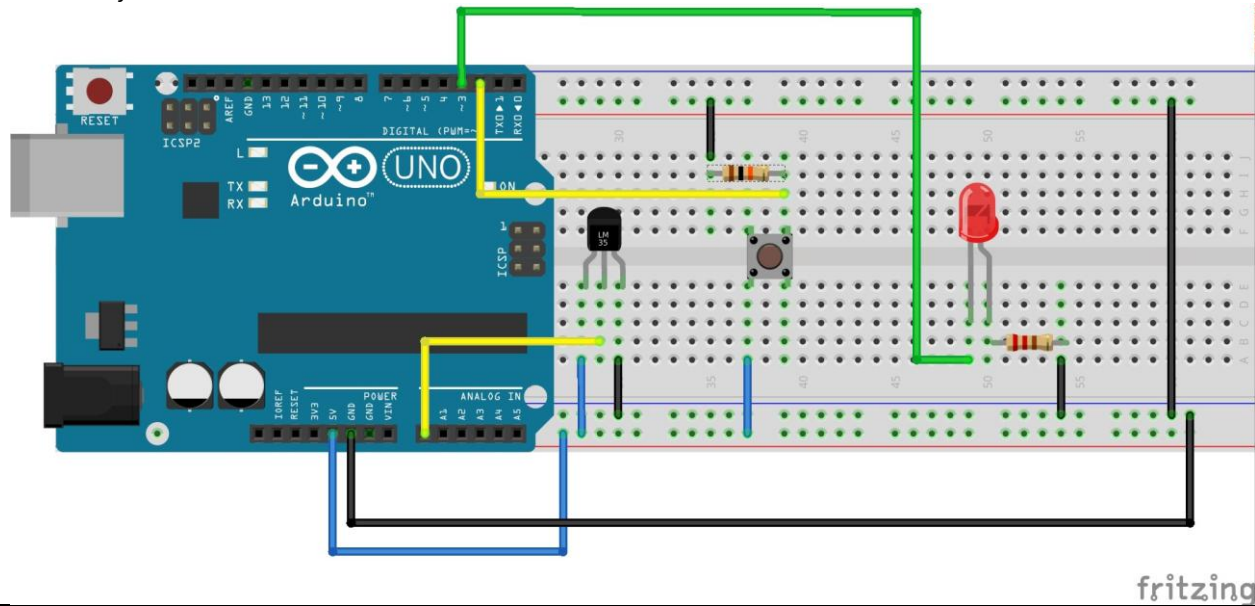
Los alumnos deberán introducir en el código la cantidad de agua (en gramos) y el la cantidad de sal (en mol) en cada experiencia, valorándose que incluyan el equivalente en agua del calorímetro.

Hay que tener especial cuidado con el sensor de temperatura, ya que los pines han de colocarse de manera específica:

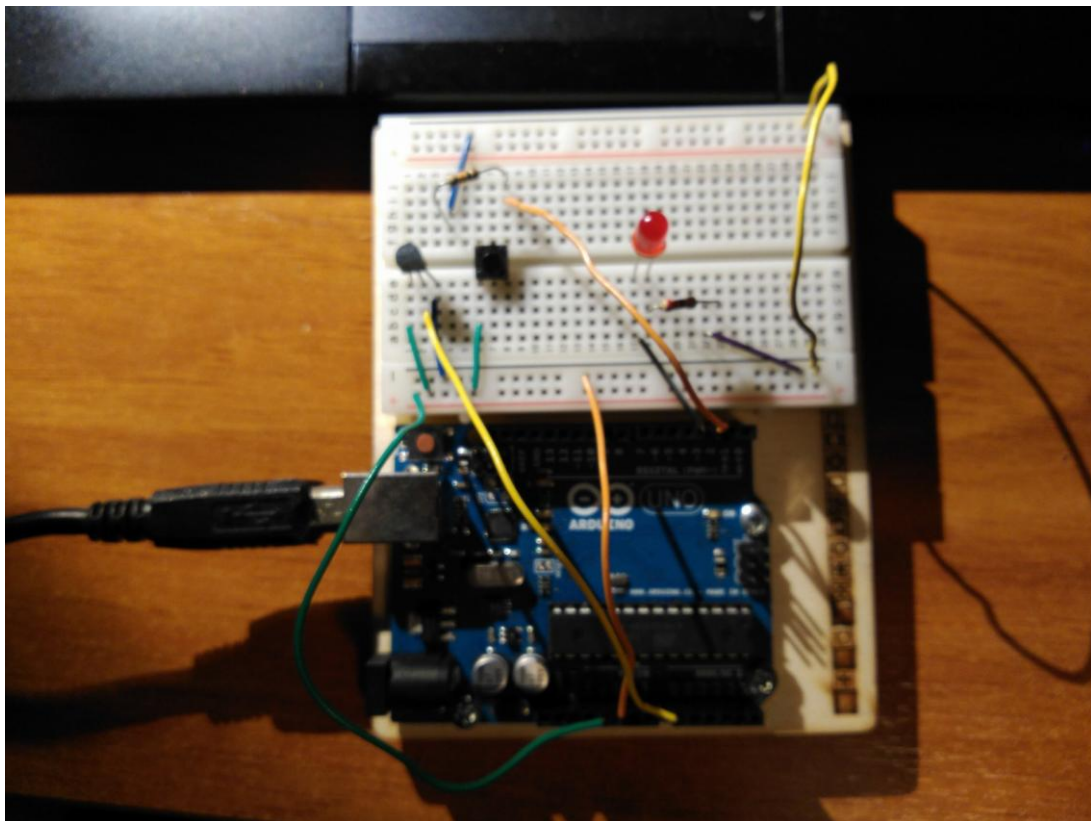
- Con la leyenda del sensor mirando hacia uno,

- \* Pin izquierdo: a 5V
- \* Pin derecho: a GND
- \* Pin central: a puerto A0

### Diseño del montaje



### Foto del montaje



### Código

```
/** Included libraries */  
/** Global variables and function definition */  
const int sensorPin = A0; // pin del sensor de temperatura  
const int ledPin = 9; // pin del led de calibracion  
int sensorValor = 0; // valor inicial del sensor  
int sensorActual = 0; // valor mínimo del sensor  
float tempInicial = 0; // vaoor de la tempetatura del agua al comienzo de la práctica  
float tempActual = 0; // valor de la temperatura duarante el calculo  
float salida = 0;  
float entalpia = 0;  
float masa_agua = 100; // VALOR QUE EL ALUMNO HA DE CAMBIAR SEGUN LA  
CANTIDAD DE GRAMOS DE AGUA INTRODUCIDOS  
float moles = 0.1; // VALOR QUE EL ALUMNO HA DE CAMBIAR EN FUNCION DEL Nº  
MOLES DE SAL INTRODUCIDOS  
const int led_rojo = 3;  
const int boton_0 = 2;  
  
void setup() {  
// La luz del led 13 se enciende durante el calibrado  
pinMode(13, OUTPUT);  
digitalWrite(13, HIGH); // Se enciende el led de la placa durante la toma de  
temperatura inicial  
Serial.begin(9600); //inicializo el puerto serie  
pinMode(boton_0, INPUT); // establezco el pin digital 2 para el botón  
pinMode(led_rojo, OUTPUT); // establezco el pin digital 3 para el led  
// para calcular la temperatura inicial espero durante 12000 ms (12 s) para llegar a  
la temperatura de equilibrio  
// y saber cual es la temperatura inicial  
// con la función millis() el sensor sigue funcionando durante el tiempo indicado  
while (millis() < 12000) {  
sensorActual = analogRead(sensorPin);  
sensorValor = sensorActual;  
}  
// cambio la lectura del sensor a grados Celsius  
tempInicial = (((sensorValor * 5.0 / 1024.0) - .5) * 100);
```

```
Serial.print("Temperatura inicial");
Serial.print("\t");
Serial.print(tempInicial);
Serial.println("");
// señal de fin de toma de temperatura inicial. Se apaga el led de la placa
digitalWrite(13, LOW);
Serial.print("Pulsa el botón cuando hayas introducido la sal en el calorímetro y
mantenlo pulsado hasta acabar");
Serial.println("");
Serial.print("Para saber si has acabado, mira en el Serial Monitor que el incremento
de temperatura se mantenga en cero");
Serial.println("");
}
/** Loop */
void loop() {
float Boton = digitalRead(boton_0);
if (Boton == 0) {
digitalWrite(led_rojo, HIGH);
delay(200);
digitalWrite(led_rojo, LOW);
delay(100);
} else {
// Serial.print("A partir de ahora, agita suavemente la disolución. MANTEN
PULSADO EL BOTÓN");
sensorActual = analogRead(sensorPin);
sensorValor = sensorActual;
tempActual = (((sensorValor * 5.0 / 1024.0) - .5) * 100);
//Serial.print("Temperatura actual");
//Serial.print("\t");
//Serial.print(tempActual);
//Serial.println("");
Serial.print("El incremento de T(°C) ");
Serial.print("\t");
salida = tempActual - tempInicial;
Serial.print("\t");
Serial.print(salida);
```

```
Serial.print("\t");  
Serial.print("Entalpía ");  
Serial.print("\t");  
entalpia = masa_agua * salida / moles;  
Serial.print(entalpia);  
Serial.println("");  
}  
}
```

*Comentarios*

## GUÍA DIDÁCTICA

**ASIGNATURA(S):** FÍSICA Y QUÍMICA / TÉCNICAS DE LABORATORIO

**NIVEL EDUCATIVO:** ESO / BACHILLERATO

**DESCRIPCIÓN DEL PROYECTO:** Cálculo de las entalpías de disolución de diversos solutos mediante el empleo de una placa Arduino en el laboratorio de Ciencias.

**OBJETIVOS:**

- Aplicar los conocimientos de programación adquiridos.
- Comprender que las disoluciones, al igual que todas las reacciones, son exotérmicas o endotérmicas.
- Aplicar los conocimientos teóricos a situaciones reales.

**COMPETENCIAS GENÉRICAS:**

- Científica
- Matemática
- Tecnológica

**RETO O PROBLEMA REAL QUE TIENEN QUE RESOLVER LOS ALUMNOS:**

Calcular las entalpías de disolución de distintos solutos en el laboratorio comparándolos con los datos tabulados.

**TEMPORALIZACIÓN Y FASES DEL PROYECTO:**

4 horas

- 1 h diseño del código
- 1 h prueba en el laboratorio y corrección del código
- 1 h realización de la experiencia
- 1 h realización del informe y esquemas oportunos

**RECURSOS (componentes, otros materiales,...):**

- PC
- Placa Arduino
- Led
- Pulsador
- Resistencias
- Cables
- Materiales propios del laboratorio

**ATENCIÓN A LA DIVERSIDAD (diferentes niveles de dificultad y retos de ampliación):**

- Empezar por una medida simple, sin cálculo de entalpía por el programa.
- Modificar la salida con entalpías.
- Realización sin programar, con termómetro analógico.

**EVALUACIÓN:**

**PROPUESTA DE ROLES DEL EQUIPO DEL PROYECTO Y FUNCIONES (diseñador, programador, gestor de recursos, responsable de la memoria,...) :**

- Programador
- Gestor de recursos Arduino
- Gestor de recursos de laboratorio
- Memoria conjunta: Trabajo colaborativo con Google Drive (Docs, Presentaciones)



### RÚBRICA DE EVALUACIÓN:

	Aspectos	Sobresaliente	Bueno	Aceptable	No aceptable	%
		4	3	2	1	
Pensamiento computacional y programación	<b>Estudio teórico</b>	Consiguen que el programa realice los cálculos correctamente, teniendo en cuenta el equivalente en agua del calorímetro	Consiguen que el programa realice los cálculos correctamente	No consiguen que los cálculos los realice el programa pero saben hacerlo a mano.	No saben realizar los cálculos	40
	<b>Presentación</b>	Presentación correcta sin faltas de ortografía, documentada con imágenes y capturas de pantalla del código. Incluyen los cálculos matemáticos, capturas de resultados y hacen comparativa con datos tabulados. Incluyen bibliografía.	Presentación correcta sin faltas de ortografía, documentada con imágenes y capturas de pantalla del código. Incluyen los cálculos matemáticos, Incluyen bibliografía.	Presentación escasa, sin cálculos y/o faltas de ortografía.	Presentación incompleta sin cálculos.	20
	<b>Código</b>	Presenta un código bien documentado, con variables reconocibles que soluciona el problema propuesto. Además, presenta modificaciones de mejora para otros posibles problemas (cálculo de pH...).	Presenta un código correcto y bien documentado que soluciona el problema propuesto.	Presenta un código que, aun correcto y que soluciona el problema propuesto, no es claro y no está bien documentado.	Presenta un código incompleto que no soluciona el problema planteado.	40