



**Dirección General de Asuntos
del Personal Académico**
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Proyecto PAPIME PE106816

PRÁCTICAS DE FÍSICA DESARROLLANDO TU PROPIO INSTRUMENTO ELECTRÓNICO DE MEDICIÓN

Coordinación

Miguel Ángel Bañuelos Saucedo ICAT

Revisión técnica

Eduardo José Vega Murguía ICAT

Participantes

Sergio Alejandro Carrillo Araujo	CCH-Sur
Ingrid Escobedo Estrada	F. de Química
Javier Fonseca Madrigal	CCH-Sur
Emilio García Valdez	CCH-Sur
David León Salinas	ENP 7
Milagros Pacheco Castañeda	ENP 5
Ismael Rivera Jiménez	CCH-Sur
Gonzalo Victor Rojas Cárdenas	CCH-Oriente
Eduardo José Vega Murguía	ICAT

2019



Esta obra está licenciada bajo la Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional.

Para ver una copia de esta licencia, visite:

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

O envíe una carta a:

Creative Commons
PO Box 1866
Mountain View, CA 94042, USA.

ÍNDICE

Presentación.....	4
1. TIEMPO DE RESPUESTA DE UN SER HUMANO.....	6
2. MOVIMIENTO RECTILÍNEO	21
3. MODELO “CASERO” DEL PROCESO DE RESPIRACIÓN	44
4. EQUILIBRIO TÉRMICO	55
5. TRANSFERENCIA DE CALOR	66
6. PRESIÓN ATMOSFÉRICA Y ALTURA	76
7. CURVAS DE CALENTAMIENTO Y ENFRIAMIENTO DE LÍQUIDOS CON SENSORES EN ARDUINO..	89
8. MEDICIÓN DE DISTANCIA CON MOVIMIENTO ROTACIONAL.....	112
9. DETERMINACIÓN DE ÍNDICE ULTRAVIOLETA CON ARDUINO	129
10. MIDIENDO LA TEMPERATURA EMPLEANDO EL ARDUINO PARA AYUDAR A IDENTIFICAR PAREDES AISLANTES Y DIATÉRMICAS	142
11. SENSOR ULTRASÓNICO PARA INVIDENTES	154
12. LEY DE BOYLE.....	163

Presentación

La tecnología se ha vuelto un objeto de uso cotidiano. Vivimos rodeados de teléfonos inteligentes, pantallas de televisión inteligentes, tabletas electrónicas, etc. Es evidente que los alcances de la tecnología han dejado de sorprendernos, pero el reto es no sólo ser usuarios de la tecnología, sino creadores de ella.

El manejo, dominio y entendimiento de la tecnología para crear y resolver problemas puede cultivarse desde ambientes pre-universitarios y como ejemplo están los grupos de estudiantes que ya desde ese nivel son capaces de construir robots y participar exitosamente en concursos internacionales. Podemos estar orgullosos de ellos, pero no debemos conformarnos.

La física siempre ha estado presente en los planes de estudio de nivel medio superior, y para apoyar su enseñanza en la UNAM se ha invertido en instalaciones y equipo, con resultados positivos.

El presente proyecto ha tenido los objetivos fundamentales siguientes:

En primer lugar, generar una plataforma (talleres material didáctico, blog de discusión y otros) para actualizar a los profesores en el manejo de dispositivos electrónicos (módulos basados en microcontrolador, programación, sensores, adquisición de datos desde una computadora personal, etc.) que les permitan no solo diseñar una práctica de laboratorio, sino su propio instrumento de medición. El manejo de estas tecnologías permitirá a los profesores desarrollar sus propias prácticas, en función de sus propios objetivos didácticos, y de los materiales disponibles. El objetivo parece muy ambicioso, pero se basa en la utilización de módulos Arduino, que tienen un lenguaje de programación sencillo, y que cuentan con un gran número de bibliotecas y módulos de sensores disponibles en el mercado nacional.

En segundo lugar, las prácticas desarrolladas tienen énfasis en el bajo costo de los componentes empleados, para facilitar su uso en los planteles del bachillerato universitario. Se ha incluido la información necesaria (programas, listas de partes, etc.) que permiten su fácil reproducción y modificación o reparación. Si se descompone un componente, resulta sencillo reemplazarlo.

Por último, se busca que los estudiantes, asesorados por sus profesores, desarrollen también la capacidad no sólo de entender el fenómeno físico sino de manejar las herramientas para construir el instrumento con el cual lo van a analizar. Los jóvenes de manera natural interactúan con dispositivos electrónicos, y el acercarlos a sus principios de operación puede hacer más atractiva para ellos la actividad experimental. Algunos de ellos podrán comenzar una carrera en áreas tecnológicas, y este tipo de experiencias les servirá de antecedente fundamental. Los que opten por otro tipo de carreras, tal vez aprendan a valorar la importancia del desarrollo tecnológico.

Prácticas de física desarrollando tu propio instrumento electrónico de medición

La presente propuesta, surge como una extensión del curso homónimo que dentro del Programa de Actualización y Superación Docente (DGAPA) impartí en junio de 2015, con una duración de 20 horas, y con la asistencia de 17 profesores del bachillerato de la UNAM. Durante el curso quedó de manifiesto el interés de los participantes por construir su propio instrumento, que satisfaga sus propuestas didácticas. Se busca generar una estructura que dé un mejor apoyo a la actualización de los profesores en el campo de la instrumentación electrónica aplicada a experimentos de física.

Miguel Ángel Bañuelos Saucedo
Instituto de Ciencias Aplicadas y Tecnología
Universidad Nacional Autónoma de México
2019

1. Tiempo de respuesta de un ser humano

1. TIEMPO DE RESPUESTA DE UN SER HUMANO

Responsable: Eduardo José Vega Murguía ICAT	Participantes:	Revisión: Emilio García Valdez CCH-Sur
---	----------------	--

Ubicación curricular

Colegio de Ciencias y Humanidades

Física 1

Unidad 1. Introducción a la Física

Escuela Nacional Preparatoria

Física 4 área 2

Unidad 1. Física de la visión y la audición

Unidad 2. Fluidos y pulsos eléctricos en el cuerpo humano

INTRODUCCIÓN

El cuerpo humano se puede considerar un sistema físico que, como cualquier otro, también responde a los estímulos que recibe con alguna respuesta que no es instantánea, sino que hay un tiempo “t” determinado que tarda en responder. Por ejemplo, cuando una persona va manejando un vehículo y de repente ve que se enciende la luz roja del semáforo indicando parar, no aplica inmediatamente el freno, sino tarda un tiempo en hacerlo. Este tiempo no solo corresponde al tiempo de mover el pie hacia el pedal de freno, sino también se agrega el tiempo que tarda en reaccionar el cuerpo desde el momento en percibir la luz roja hasta que el pie se empieza a mover hacia el pedal.

En este experimento se trata de determinar cuánto tiempo tarda en responder un alumno a un estímulo luminoso.

Al intentar repetir varias veces seguidas esta medida, generalmente se aprecia que no es reproducible, es decir, estas magnitudes no se pueden determinar con una sola medida. Esto es debido a que hay diversos factores que le influyen en cada ocasión, por lo que se requieren tomar varias medidas “t_i” y obtener un promedio “T”, el cual se puede calcular con la expresión:

$$T = (t_1 + t_2 + \dots + t_n) / n$$

siendo “n” el número de repeticiones de la toma de la medida. La determinación de su intervalo de incertidumbre “Δt” se puede hacer de varias maneras, según convenga a los intereses del experimento o intenciones de la clase. Una opción es considerar la mitad del intervalo al que pertenecen los datos tomando las medidas: menor “t_{min}” y mayor “t_{max}”, en este caso esto se calcularía por la expresión:

$$\Delta t \approx (t_{\max} - t_{\min}) / 2$$

1. Tiempo de respuesta de un ser humano

Otra opción es tomar el valor absoluto de la diferencia mayor entre el valor promedio y estas medidas, lo cual se obtendría de la siguiente manera:

$$\Delta t \approx \text{el mayor de } \{ |T - t_{\max}| \text{ y } |T - t_{\min}| \}$$

donde las magnitudes entre las líneas verticales indican que se trata de los valores absolutos de las diferencias. La tercera opción es calcular el promedio de los valores absolutos de las desviaciones de cada medida respecto al valor promedio, lo cual se calcula con la expresión siguiente:

$$\Delta t \approx (|T - t_1| + |T - t_2| + \dots + |T - t_n|) / n$$

Y la más común es calcular la desviación estándar o típica con la expresión:

$$\Delta t \approx ([(T - t_1)^2 + (T - t_2)^2 + \dots + (T - t_n)^2] / n)^{1/2}$$

La medida del tiempo de respuesta puede tener diversas implicaciones, por ejemplo, en la situación mencionada arriba, al frenar el vehículo recorrerá una distancia antes de parar, determinada por el tiempo de respuesta, la velocidad con que avanzaba el vehículo y la fuerza resultante que se le aplica al frenar.

GUÍA PARA EL PROFESOR: El profesor elige con sus alumnos o determina cuál es el procedimiento para determinar la incertidumbre del tiempo de reacción, teniendo en cuenta que los otros métodos también pueden ser válidos (Baird, 1999).

OBJETIVOS

1) Experimental:

Medir el tiempo de respuesta de una persona ante una señal.

2) Enseñanza:

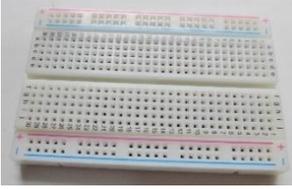
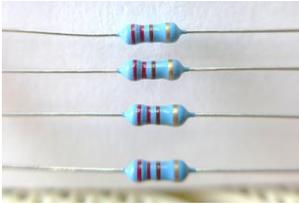
Que el alumno se reconozca como un sistema que tiene un tiempo de reacción para ejecutar una tarea que se le solicita al momento de recibir una señal.

Que el alumno se percate que hay magnitudes físicas que para determinar su valor se requiere repetir varias veces las pruebas ya que no son reproducibles para que, con base en un criterio estadístico, asignarle un valor a la magnitud que se desea medir.

MATERIAL

GUÍA PARA EL PROFESOR: El profesor podrá explicar a los alumnos que: Se propone hacer la actividad con cualquiera de dos posibles Sketchs o programas del Arduino para lograr sus objetivos. Estos programas se presentan respectivamente en las secciones "A" y "B" del anexo 1. El material básico indispensable que se requiere para realizar la actividad con cualquiera de los programas es el mismo; sin embargo, hay algunos componentes que se requieren para uno que no lo son para el otro y viceversa. Los recursos materiales exclusivos que se requieren con cada uno de estos programas se denotan con la letra "A" o "B" en el cuadro de observaciones de la siguiente lista de material, según sean para aplicar el programa de la sección "A" o de la sección "B" del anexo. El material básico indispensable no tiene indicación alguna al respecto.

1. Tiempo de respuesta de un ser humano

Descripción	Imagen	Comentarios
<p><i>Arduino UNO</i></p>		
<p>Fotointerruptor reflexivo CNY70</p>		
<p>LEDs: Rojo Verde</p>		<p>El LED rojo es para producir la señal de estímulos. El LED verde es para indicar que se registró el tiempo de respuesta.</p>
<p>Interruptor mecánico tipo: <i>Push button</i></p>		<p>El botón es para activar el inicio de lectura del cronómetro.</p>
<p><i>Protoboard</i> 400 contactos</p>		<p>Donde se arma el circuito que se conecta al <i>Arduino</i>. El <i>Protoboard</i> se puede reemplazar por una tarjeta de impreso.</p>
<p>Resistencias: 4 de 220 ohm, 1 de 1 kohm y 3 de 10 kohm</p>		<p>Las bandas de colores para las resistencias son: 220 ohm: Rojo – Rojo – Café, 1 kohm: Café - Negro – Rojo, y 10 kohm: Café - Negro – Naranja.</p>
<p>Cable jumper macho-macho</p>		<p>De diversos tamaños y colores.</p>

1. Tiempo de respuesta de un ser humano

Descripción	Imagen	Comentarios
Computadora		<p>Puede ser de escritorio o tipo laptop.</p> <p>Se requiere con el programa "A" para conectarle el <i>Arduino</i> y transferir los datos.</p> <p>Se requiere con ambos programas, "A" y "B", para hacer el análisis de datos.</p>
Cable USB A macho a B macho		<p>Con el programa "A" para la comunicación entre la computadora y el <i>Arduino</i>.</p> <p>Se puede emplear con el programa "B" si se conecta una pila de recarga al <i>Arduino</i>.</p>
Pantalla de cristal líquido de 2 líneas con 16 caracteres		<p>Se requiere con el programa "B" para el despliegue de valores medidos. Los pines son soldados por el usuario.</p>
Interruptor mecánico tipo: Un Polo Un Tiro		<p>Se puede requerir con el programa "B" si se usan pilas como fuente de energía y para encender el <i>Arduino</i>.</p>
Pilas con portapilas y conector, o eliminador de baterías		<p>Con el programa "B". Fuente de poder de CD con pilas o eliminador de baterías. Alimentación entre 5 y 9 V.</p>

Software

GUÍA PARA EL PROFESOR: El profesor debe proporcionar a los alumnos el programa que vayan a utilizar para que lo carguen al microprocesador de Arduino, o previamente cargarlo a éste. Para la elaboración de los sketches que se proponen se consultó el Tutorial sobre Arduino (Bañuelos, 2017). El profesor puede revisar en el sitio oficial del Arduino otro ejemplo de programa para medir el tiempo de reacción diferente al que se propone aquí (Desai, 2016).

1. Tiempo de respuesta de un ser humano

- Arduino IDE 1.8.2 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>
- **"A"**. *Sketch* de la sección "A" del anexo 1. Programa con presentación de resultados en la pantalla de la computadora con aplicación: "CoolTerm". Al hacer la actividad es necesario mantener continuamente conectado el Arduino a la computadora para capturar los datos.
- **"B"**. *Sketch* de la sección "B" del anexo 1. Programa para medir el Tiempo de Respuesta de una persona desplegando los resultados en la pantalla de Cristal líquido con la aplicación "LiquidCrystal.h". En este caso, la actividad se puede hacer sin que la tarjeta Arduino esté conectada a la computadora.
- Programa CoolTerm 1.4.7 o superior. Disponible en: <http://freeware.the-meiers.org/>.
- Excel o Quatro Pro o cualquier otro programa de hoja de cálculo. Al final lo recomendable es elaborar el histograma de datos con algún programa de hoja de cálculo.

DESARROLLO

1.- La actividad la deben de realizar todos los alumnos organizados en parejas. Uno activa el cronómetro mientras el otro lo para. Después se intercambian.

GUÍA PARA EL PROFESOR: El diagrama de bloque, que se presenta en la figura 1, lo puede aprovechar el profesor en clase para ejemplificar la constitución y operación del cronómetro en lugar de que los alumnos lean su explicación, por ejemplo: puede proyectar el diagrama y explicarlo verbalmente.

2.- El cronómetro para medir el tiempo de respuesta es básicamente el mismo, independientemente del programa o sketch que se emplee para medirlo. De manera general, el diagrama de bloques del cronómetro se esquematiza en la figura 1. Destacan el bloque de procesamiento de datos "ARDUINO", que controla todo el proceso; el bloque de activación del cronómetro "CONTACTO ELÉCTRICO", que da una señal al ARDUINO para producir el estímulo al que debe responder el sujeto de prueba al encender el led del bloque "LED DE SEÑAL". Cuando el sujeto ha respondido, el bloque de paro de cronómetro "OPTOINTERRUPTOR" manda la señal al ARDUINO, el cual, despliega el tiempo medido en cada prueba en el bloque de PANTALLA. Este bloque representa a la computadora, si se usa el programa "CoolTerm", o a la pantalla de cristal líquido (PCL) si se emplea la biblioteca "liquidCrystal.h".

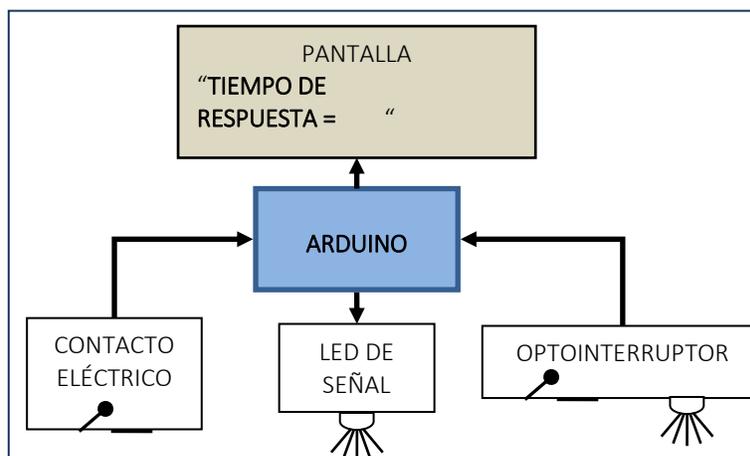


Figura 1. Diagrama de bloques del cronómetro para medir el tiempo de respuesta.

1. Tiempo de respuesta de un ser humano

3. El cronómetro se arma conforme al esquema de la figura 2. Si se va a usar la computadora para registrar los datos con el programa *CoolTerm*, se arma el circuito de la izquierda de la figura. Pero si se usa la pantalla de cristal líquido, entonces se arma del circuito de la derecha. Ambos circuitos se pueden ver más claramente en las secciones A y B del Anexo 2. Para una descripción del uso del programa *CoolTerm* se puede consultar el Tutorial sobre Arduino (Bañuelos, 2017).

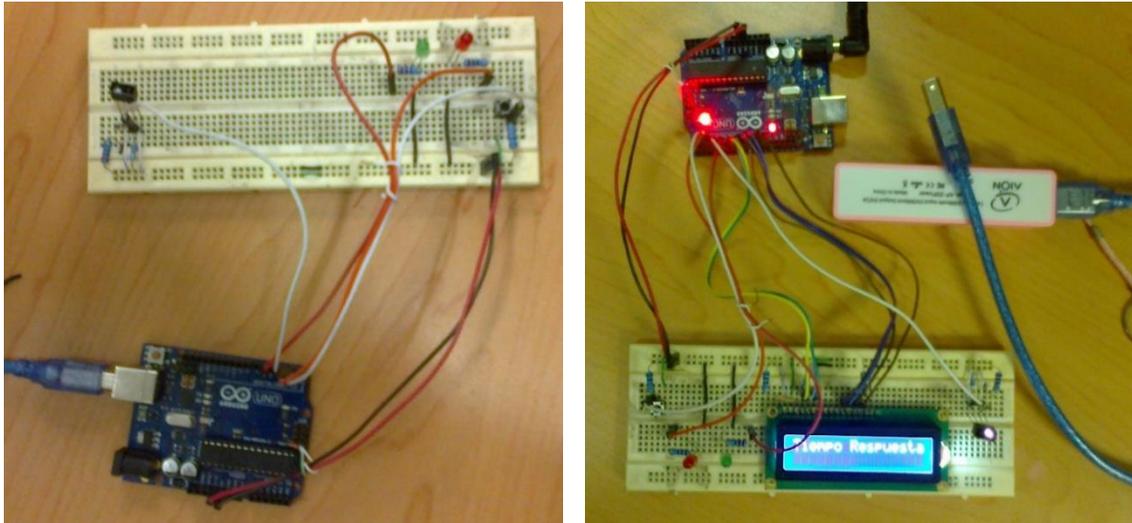


Figura 2. Cronómetro para medir el tiempo de respuesta. A la izquierda, el diagrama cuando se usa el *CoolTerm*. A la derecha, el diagrama cuando se usa la biblioteca “*LiquidCrystal.h*”.

Cuando se enciende el reloj, se activa la pantalla donde aparece el título de la actividad “Tiempo Respuesta” o “T Respuesta”.

4. Se selecciona al alumno que realizará la prueba y a quién se le medirá el tiempo de respuesta. Este debe poner su mano junto al interruptor optorreflexivo. Mientras que otro alumno se prepara para activar el inicio de la toma de datos y encender la luz estímulo con el interruptor mecánico.

El segundo alumno oprimirá el botón en el momento que lo deseé y de manera que no se dé cuenta el primer alumno. En el momento que se oprima el botón, se encenderá la señal de estímulo que corresponde a un led de color y se activará el cronómetro. Entonces el alumno evaluado deberá poner su mano sobre el interruptor optorreflexivo para parar el conteo del cronómetro, apagar el led de señal de estímulo y tomar el tiempo de respuesta. En ese momento se presentará en la pantalla el tiempo de respuesta “t” medido en milisegundos.

Se repite la prueba el número de veces que se haya determinado, se procede a calcular el valor promedio y el parámetro de incertidumbre de la medición.

RESULTADOS

5. Los tiempos de respuesta se pueden capturar en el programa *CoolTerm* en formato “*txt*” para después registrarlos en la tabla 1. Ésta, además contendrá el intervalo de incertidumbre que se haya determinado o, si se usa la pantalla de cristal líquido donde se muestran los resultados, estos se van

1. Tiempo de respuesta de un ser humano

registrando aparte para después pasarlos a la tabla. La tabla puede contener los resultados de varios estudiantes, por ejemplo, de los miembros del equipo de trabajo.

Tabla 1. Tiempos de respuesta de los estudiantes.

Medida [ms]	Estudiante 1	Estudiante 2	Estudiante3
t_1	579	1885	
t_2	449	501	
t_3	401	254	
.	351		
.	.		
.	.		
t_n	345		
Promedio T	351.5		
Intervalo de incertidumbre determinada como Desviación estándar	90.1		

Es importante indicar en la tabla la forma como se determina el intervalo de incertidumbre. El tiempo de respuesta se mide en milisegundos, ya que es la escala de tiempo con que opera el programa del Arduino. Por lo tanto, también los promedios e intervalos de incertidumbre registrados en la tabla estarán en milisegundos.

La tabla puede escribirse directamente en una hoja de cálculo, como *Excel* o *Quatro Pro*, o abrirse en ella si se guardaron con el programa *CoolTermWin*, donde se faciliten los cálculos y se pueda obtener un histograma, como el que se muestra en la figura 3.

GUÍA PARA EL PROFESOR: El profesor debe indicar como proceder a hacer el histograma y a determinar el valor de los intervalos de incertidumbre según el criterio que hayan adoptado.

1. Tiempo de respuesta de un ser humano

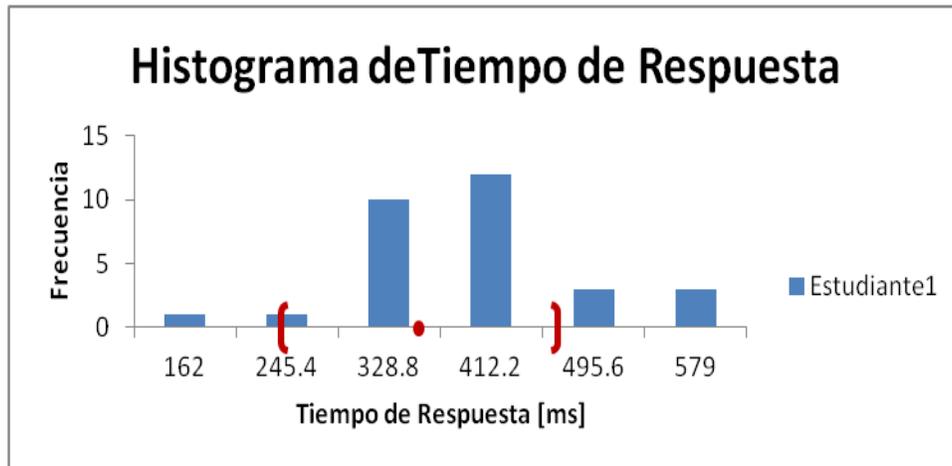


Figura 3. Ejemplo de histograma de un estudiante, con 30 datos. El punto rojo indica el valor del promedio y los paréntesis los extremos de una desviación estándar respecto al promedio.

DISCUSIÓN Y CONCLUSIONES

GUÍA PARA EL PROFESOR: Para generar una discusión y análisis en este sentido, antes de comentar el significado de los resultados se pueden comentar o no los párrafos que siguen con los alumnos y presentar las preguntas enlistadas.

6. Esta actividad permite entender que el cuerpo humano, aunque es un organismo biológico, también puede describirse como un sistema físico con propiedades o magnitudes que no pueden determinarse con una sola medición y que además, pueden cambiar debido a que no corresponde a una propiedad constante ni determinada por un solo factor, sino que puede ser afectada por múltiples causas que al azar confluyen en cada prueba. Por eso la necesidad de repetir la prueba varias veces para obtener un promedio, que representará el conjunto de datos obtenido y solo nos da un valor aproximado para expresar la magnitud medida.

Al repetir la medición del tiempo de reacción se obtienen lecturas distintas, por lo que es conveniente que este valor se exprese dentro de un intervalo que consideraríamos de incertidumbre, o dentro del cual puede obtenerse un resultado con mayor probabilidad, sin quitar la posibilidad de obtener un resultado mayor o menor a dicho intervalo en algún otro momento cuando se intente medir de nuevo el tiempo de respuesta a través de un promedio.

- ¿Cuáles creen ustedes que son las razones por las que se les pide repetir la prueba varias veces y se tomaron varias medidas?
- ¿Por qué las diversas medidas no son iguales?
- ¿Por qué se considera adecuado usar el promedio como medida aproximada del tiempo de reacción, por qué no otro resultado como, por ejemplo: el valor más frecuente o el valor medio entre el más pequeño y el más grande (el que está al centro del intervalo de distribución)?
- Si se repitiera la prueba en otra ocasión, ¿se obtendría el mismo resultado promedio?
- ¿Qué probabilidad hay de obtener otro resultado promedio que sea cercano o lejano al obtenido?

1. Tiempo de respuesta de un ser humano

- ¿Cómo se podría determinar un intervalo alrededor del promedio, dentro del cual pueda encontrarse con mayor probabilidad un nuevo promedio obtenido al repetir la prueba en otra ocasión?
- Si se comparan los resultados de varios estudiantes entre sí, ¿puede decirse que son iguales o diferentes? ¿Por qué?

GUÍA PARA EL PROFESOR: Para orientar una conclusión se proponen las siguientes preguntas

- ¿Qué se podría decir respecto a la siguiente aseveración: “Las personas podemos responder ante un estímulo inmediatamente”?
- Con su equipo generen una conclusión sobre lo que significa: “Cualquier ser humano tiene un tiempo de respuesta ante un estímulo”.

REFERENCIAS

Baird, D. C. (1991). Experimentación: Una introducción a la teoría de mediciones y al diseño experimental. Segunda edición. México: Prentice – Hall Hispanoamericana S. A.

Bañuelos, M. A. (2017). Tutorial sobre Arduino. México: ICAT – UNAM.

Desai, J. (2016). Measure your reaction time c GPL3+. Arduino. Project HUB. Recuperado el 04 de junio de 2019 de https://create.arduino.cc/projecthub/Blue_jack/measure-your-reaction-time-d68f90.

1. Tiempo de respuesta de un ser humano

ANEXOS

ANEXO 1. Programas de Arduino

Se proponen dos formas de programar el Arduino para medir el tiempo de reacción, cada uno se distingue por el dispositivo para desplegar los resultados medidos. El primero se basa en usar la aplicación “CoolTermWin” que permite mostrar los resultados en la pantalla de la computadora en tiempo real para después guardarlos en formato “txt”, y abrirlos para procesarlos y analizarlos en el programa de Hoja de Cálculo. El segundo se basa en la aplicación “#include <LiquidCrystal.h>” que permite desplegar los resultados en la pantalla de cristal líquido, los cuales deben de ser capturados al momento para después cargarlos al programa de Hoja de Cálculo. Sin embargo, ambos se basan en el programa que permite usar el Arduino como cronómetro que se encuentra en el Tutorial sobre Arduino (Bañuelos, 2017).

Programa con presentación de resultados en la pantalla de la computadora con aplicación: “CoolTerm”.

```
/* Programa para medir el Tiempo de Respuesta de una persona.  
Presentando los resultados en la pantalla de la computadora con la aplicación “CoolTerm”  
*/
```

```
int const pinACTIV = 2; //Pin de interruptor que enciende la señal  
int const pinOpto2 = 4; //Pin de fotointerruptor de tiempo de respuesta  
int const pinLed1 = 3; //Pin del led de señal  
int const pinLed2 = 5; //Pin del led del fotointerruptor de tiempo de respuesta  
int valorACTIV, valorOpto2;  

```

```
void setup() {  
  // Inicialización de los pines  
  pinMode(pinACTIV, INPUT);  
  pinMode(pinOpto2, INPUT);  
  pinMode(pinLed1, OUTPUT);  
  pinMode(pinLed2, OUTPUT);  
  Serial.begin(9600); // Velocidad de comunicación serial  
}
```

```
void loop() {  
  // Programa principal  
  valorACTIV = 0;  
  while(!valorACTIV){ // Espera a que se active el interruptor de señal  
    valorACTIV = digitalRead(pinACTIV); // Lee el interruptor de señal  
  }  
  while(valorACTIV){  
    valorACTIV = digitalRead(pinACTIV); // Espera a que se desactive señal  
  }  
  tiempo1 = millis();  
  digitalWrite(pinLed1, HIGH); //Enciende LED de inicio de la prueba
```

1. Tiempo de respuesta de un ser humano

```
valorOpto2 = 0;
while(!valorOpto2){
  valorOpto2 = digitalRead(pinOpto2); // Espera a que se active Opto2
}
while(valorOpto2){
  valorOpto2 = digitalRead(pinOpto2); // Espera a que se desactive opto2
}
tiempo2 = millis();
digitalWrite(pinLed2, HIGH); // Enciende LED de fin de la prueba
digitalWrite(pinLed1, LOW); // Apaga LED de inicio de la prueba

if(tiempo2 > tiempo1)
{
  tiempo = (tiempo2 - tiempo1);
  Serial.print("Tiempo de respuesta: "); // Se envía el tiempo de respuesta
  Serial.print(tiempo);                // por el puerto serial
  Serial.println (" : ms");
  delay(5000);
  digitalWrite(pinLed1, LOW); // Apaga ambos LEDs
  digitalWrite(pinLed2, LOW);
}
}
```

1. Tiempo de respuesta de un ser humano

Programa con presentación de resultados en la pantalla de cristal líquido con aplicación: "LiquidCrystal.h".

```
/* Programa para medir el Tiempo de Respuesta de una persona.
Presentando los resultados en la pantalla de Cristal líquido con la aplicación "LiquidCrystal.h"
*/
#include <LiquidCrystal.h>
//Se definen los pines de comunicación con el LCD
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

//Los pines del reloj son:
int const pinACTIV = 2; //Pin de interruptor que enciende la señal
int const pinOpto2 = 4; //Pin de fotointerruptor de Tiempo de respuesta
int const pinLed1 = 3; //Pin del led de señal
int const pinLed2 = 5; //Pin del led del fotointerruptor de Tiempo de respuesta
int valorACTIV, valorOpto2;
unsigned long tiempo, tiempo1, tiempo2;

void setup() {
  lcd.begin(16,2); // PCL de 2 renglones y 16 caracteres c/u
  lcd.print ("Tiempo Respuesta");

  // Inicialización de los pines
  pinMode(pinACTIV, INPUT);
  pinMode(pinOpto2, INPUT);
  pinMode(pinLed1, OUTPUT);
  pinMode(pinLed2, OUTPUT);
}

void loop() {
  // Programa principal
  //Prepara PCL
  lcd.setCursor(0,1); //Selecciona el primer carácter del segundo renglón
  lcd.print(" "); //Pone en blanco el segundo renglón

  // Medida de tiempo
  valorACTIV = 0;
  while(!valorACTIV){ // Espera a que se active el interruptor de señal
    valorACTIV = digitalRead(pinACTIV); // Lee el interruptor de señal
  }
  while(valorACTIV){
    valorACTIV = digitalRead(pinACTIV); // Espera a que se desactive señal
  }
  tiempo1 = millis();
  digitalWrite(pinLed1, HIGH); // Enciende LED de inicio de la prueba
```

1. Tiempo de respuesta de un ser humano

```
valorOpto2 = 0;
while(!valorOpto2){
  valorOpto2 = digitalRead(pinOpto2); // Espera a que se active Opto2
}
while(valorOpto2){
  valorOpto2 = digitalRead(pinOpto2); // Espera a que se desactive opto2
}
tiempo2 = millis();
digitalWrite(pinLed2, HIGH); // Enciende LED de fin de la prueba
digitalWrite(pinLed1, LOW); // Apaga LED de inicio de la prueba

if(tiempo2 > tiempo1)
{
  tiempo = (tiempo2 - tiempo1);

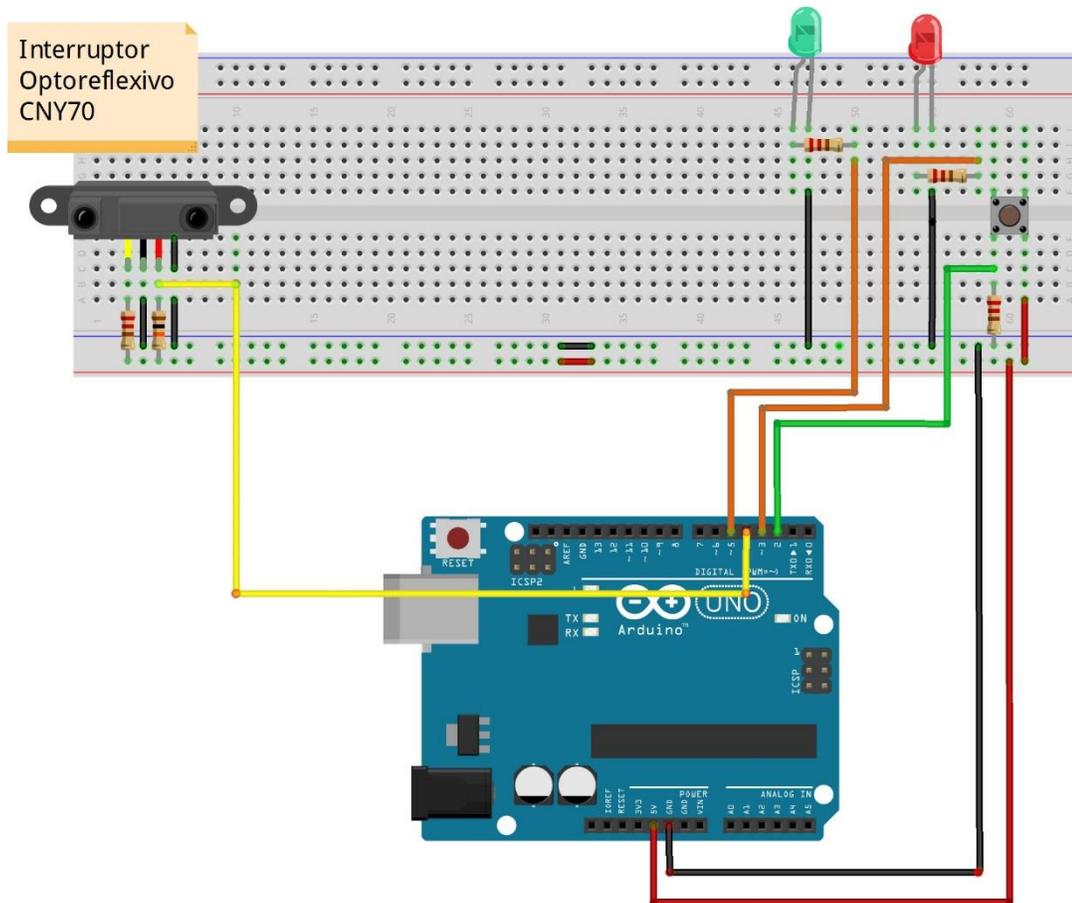
//Muestra en pantalla el tiempo de respuesta
lcd.setCursor(0,1);
  lcd.print(tiempo);
  delay(5000);

//Apaga leds
  digitalWrite(pinLed1, LOW);
  digitalWrite(pinLed2, LOW);
}
}
```

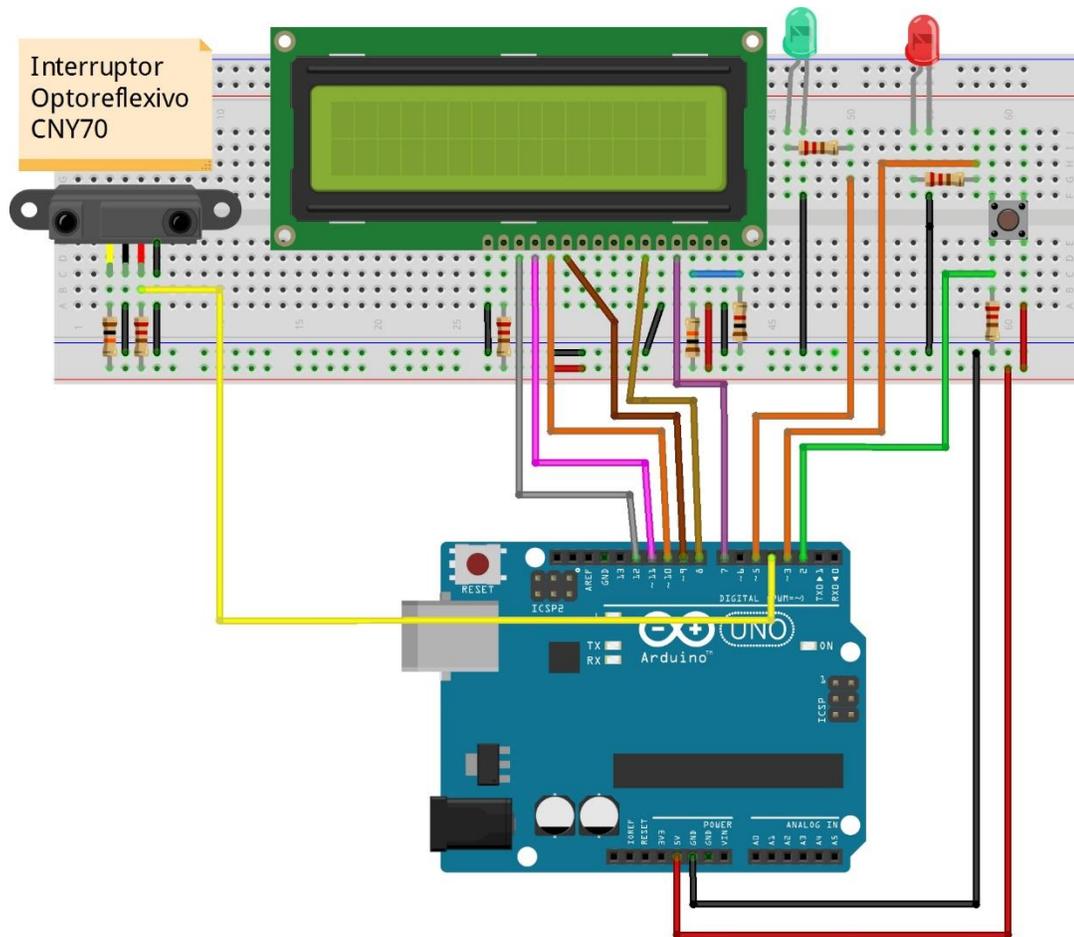
1. Tiempo de respuesta de un ser humano

ANEXO 2. Circuito electrónico

- A. Circuito para el Programa con presentación de resultados en la pantalla de la computadora con aplicación "CoolTerm".



1. Tiempo de respuesta de un ser humano
- B. Circuito para el programa con presentación de resultados en la pantalla de cristal líquido con la biblioteca "LiquidCrystal.h".



2. MOVIMIENTO RECTILÍNEO

Responsable: Ismael Rivera Jiménez CCH-SUR	Participantes: Miguel Á. Bañuelos Saucedo ICAT	Revisión: Eduardo José Vega Murguía ICAT
--	--	--

Ubicación curricular

Colegio de Ciencias y Humanidades

Física I

Unidad 2: Mecánica de la partícula: leyes de Newton

INTRODUCCIÓN

El movimiento está en todas partes, para identificarlo basta con observar los objetos a nuestro alrededor para ver, por ejemplo, como se mueven las nubes, el volar de las aves, un automóvil desplazándose por la carretera, el caminar de las personas en las competencias de atletismo, etcétera. De una manera u otra todos los objetos o cuerpos tienden a moverse, ya sea en trayectorias rectas o curvas; por ejemplo, un ciclista que se mueve en una avenida lo hace en una trayectoria recta, al igual que una piedra que cae; en un juego de baloncesto el balón lanzado a la canasta describe una trayectoria curva, de la misma manera que una piedra lanzada horizontalmente cae al piso.

Al observar estos movimientos, se pueden establecer preguntas como las siguientes: ¿de qué manera se mueven los objetos?, ¿se puede modelar este comportamiento?, ¿se puede predecir el comportamiento de un objeto cuando se mueve?

Para dar respuesta a éstas y a otras preguntas que pudieran surgir, es importante reproducir en el laboratorio estos movimientos, con experimentos controlados que permitan entender y comprender su comportamiento.

La propuesta de esta actividad experimental es precisamente ese fin, reproducir movimientos rectilíneos que corroboran los modelos teóricos. Diferenciar entre los movimientos rectilíneo uniforme y uniforme acelerado, así como hacer un análisis de estos movimientos mediante sus datos y gráficas, asociándoles el modelo matemático correspondiente.

OBJETIVOS

Objetivos de aprendizaje:

- Identificar un sistema en equilibrio.
- Reconocer las semejanzas y diferencias entre las gráficas de movimiento rectilíneo uniforme y rectilíneo uniforme acelerado.
- Analizar el tipo de movimiento que puede haber en un sistema en equilibrio.
- Analizar el movimiento que puede haber en un sistema no equilibrado.
- Identificar las características de cada movimiento.

2. Movimiento rectilíneo

Objetivos de operación:

- Manejar los programas de Arduino, CoolTerm y Excel.
- Construir un sistema mecánico.

MATERIAL

En la siguiente tabla se sugieren los materiales para el desarrollo de estas experiencias de laboratorio.

Descripción	Imagen	Comentarios
Rin de bicicleta y soporte para el rin		A este rin se le acopla un soporte en el eje de giro para que se pueda sostener mediante algún tipo de dispositivo a cierta altura. El rin fungirá como una polea.
Placas de acrílico de 30 cm por 30 cm. Tapas para tubo de PVC de 3.0 pulgadas. Dos tornillos con tuerca y rondana de 12.8 mm de largo y 3.9 mm de diámetro.		Estos materiales serán ensamblados como se presenta en la imagen de la izquierda. Este arreglo representará las masas del sistema.
Rondanas de metal		Para cambiar la masa se sugiere utilizar rondanas que se colocarán dentro de las tapas o usar cualquier otro material. Las imágenes de la izquierda nos muestran este hecho.
Hilo de nylon		Usar hilo de nylon y un par de clips, para hacer las asas de las tapas y la cuerda que pasará en el rin. La imagen izquierda muestra la idea.
Ensamblado del sistema		
Computadora		En la computadora se tienen los programas de Excel, Arduino y Coolterm para hacer el análisis de datos y en ella también se conecta la tarjeta.

2. Movimiento rectilíneo

Descripción	Imagen	Comentarios
Arduino UNO en su base de acrílico (la base es opcional).		En el Arduino se le carga el programa que controlará el sensor conectado a la computadora.
Sensor ultrasónico HC-SR04, base de acrílico para el sensor y tira de madera para base del sensor.		La imagen de la izquierda muestra el sensor colocado en su base y ésta sujeta a una tira de madera con tornillo pija.
Cables para conectar el Arduino con la computadora (Cable USB A macho a B macho) y cables para conectar el sensor y la tarjeta (jumpers macho-macho).		Se utiliza un cable USB con conectores A y B, con el cual la computadora envía y recibe la información de la tarjeta. Cuatro cables tipo jumper macho-hembra para el sensor.

Software

Para llevar a cabo este proceso experimental se requiere tener instalado en la computadora el siguiente software:

- Programa Arduino IDE 1.8.2 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>.
- Programa CoolTerm 1.4.7 o superior. Disponible en: <http://freeware.the-meiers.org/>.
- Excel y Word o equivalentes.

La figura 1 nos muestra los ambientes de los programas Arduino IDE y CoolTerm.

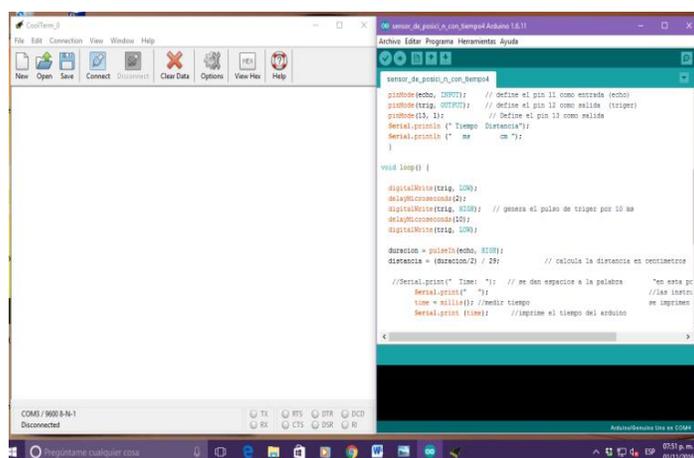


Figura 1. Ambientes de CoolTerm (izq.) y Arduino IDE (der.) instalados en la computadora.

2. Movimiento rectilíneo

DESARROLLO

Conexiones electrónicas

El sensor ultrasónico, se conectará al Arduino como se muestra en la figura 2 y se detalla en el anexo 2.

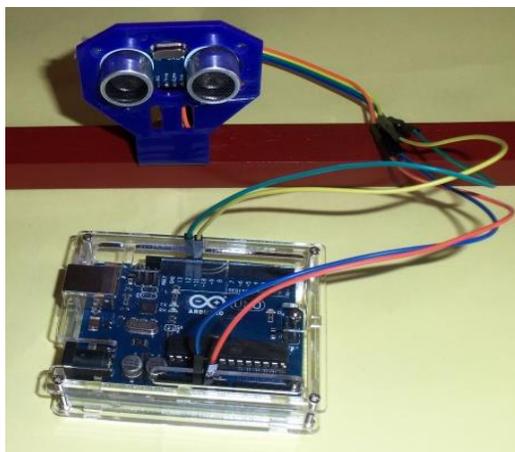


Figura 2. Conexión del sensor al Arduino.

La tarjeta Arduino se deberá conectar a la computadora utilizando el cable USB, como se ilustra en la figura 3.

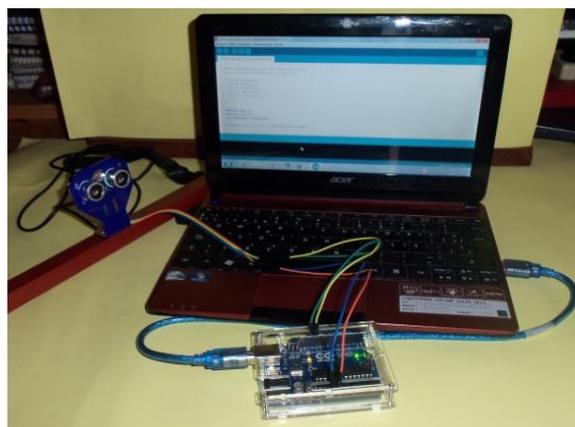


Figura 3. Conexión en la computadora.

Montaje del sistema mecánico

A las placas de acrílico de 30 x 30 cm, y a las dos tapas para tubo de PVC se les deberá hacer una pequeña perforación en el centro, para permitir su unión mediante una tuerca y un tornillo. A las tapas se les deberá realizar un par de perforaciones laterales, para permitir su sujeción con una

2. Movimiento rectilíneo

cuerda. La rueda de bicicleta se deberá sujetar al techo del laboratorio de manera que pueda girar libremente. Sobre el canal del rin, se hará pasar una cuerda en cuyos extremos se amarran las tapas con las placas dirigidas hacia abajo. Estas placas sirven como “pared” para reflejar la señal del sensor ultrasónico y de esa manera poder estimar su posición. La figura 4, muestra cómo se hace el montaje del sistema mecánico para el desarrollo de la actividad experimental.



Figura 4. Dispositivo para realizar las actividades experimentales.

Fase 1. Movimiento rectilíneo uniforme

Para realizar esta actividad experimental, se debe colocar el sensor de tal manera que éste quede por debajo de una de las masas y muy próxima a ella (entre 10 y 12 cm de separación, ya que a una distancia menor el sensor no medirá correctamente). En la figura 5 se ilustra este hecho.

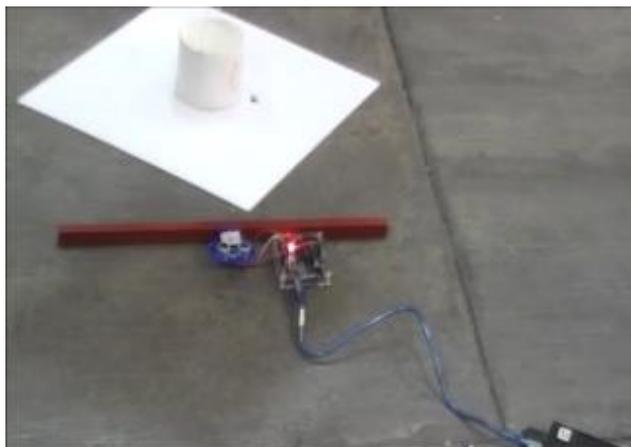


Figura 5. Sensor debajo de una de las masas.

2. Movimiento rectilíneo

Una vez conectado el Arduino y cargado con el programa (ver Anexo), este comenzará a enviar datos a la computadora sobre la posición. Debe ahora sincronizarse el inicio del registro de datos mediante el programa CoolTerm, con el inicio del movimiento. Para ello, los equipos esperarán la instrucción del profesor para que un alumno active el registro en la computadora, mientras otro inicia el movimiento de las placas. Para el caso de movimiento rectilíneo uniforme, manualmente se le imprimirá un impulso hacia arriba a la masa que está cerca del suelo, a través de la cuerda que la sostiene; de tal manera, que adquiera un movimiento en la dirección vertical como lo muestra la figura 6. La adquisición de los datos experimentales y el análisis de esta fase, se comentan en el Anexo 3.



Figura 6. Impulso sobre la cuerda para que la masa adquiera un movimiento.

Fase 2. Movimiento rectilíneo uniforme acelerado

El desarrollo de esta actividad es muy similar a la anterior. La diferencia entre ambas fases consiste, en que ahora a una de las masas se le agregan varias rondanas (ver figura 7), que representarán un cambio de esta masa y por lo tanto un cambio en el peso. Esto hará que el sistema se mueva por sí mismo, adquiriendo una aceleración. El registro de los datos mediante el programa CoolTerm, deberá sincronizarse con el instante en que se suelta la masa y comienza el movimiento. El tratamiento de los datos se menciona en el Anexo 3.

2. Movimiento rectilíneo



Figura 7. Rondanas en una de las masas.

Es conveniente en este momento hacer una observación, en cualquiera de las dos fases se puede hacer que la toma de datos se haga, ya sea de manera en que las masas se alejen del sensor, o se acerquen a éste y esto depende en cuál de ellas el sensor está colocado por debajo de ésta.

RESULTADOS

En la figura 8, se muestra en una hoja de Excel la tabla de datos y la gráfica del movimiento rectilíneo uniforme o primera fase. Se observa que hay una buena aproximación lineal, a la cual se le ajusta su línea de tendencia y ecuación, que el mismo programa puede proporcionar.

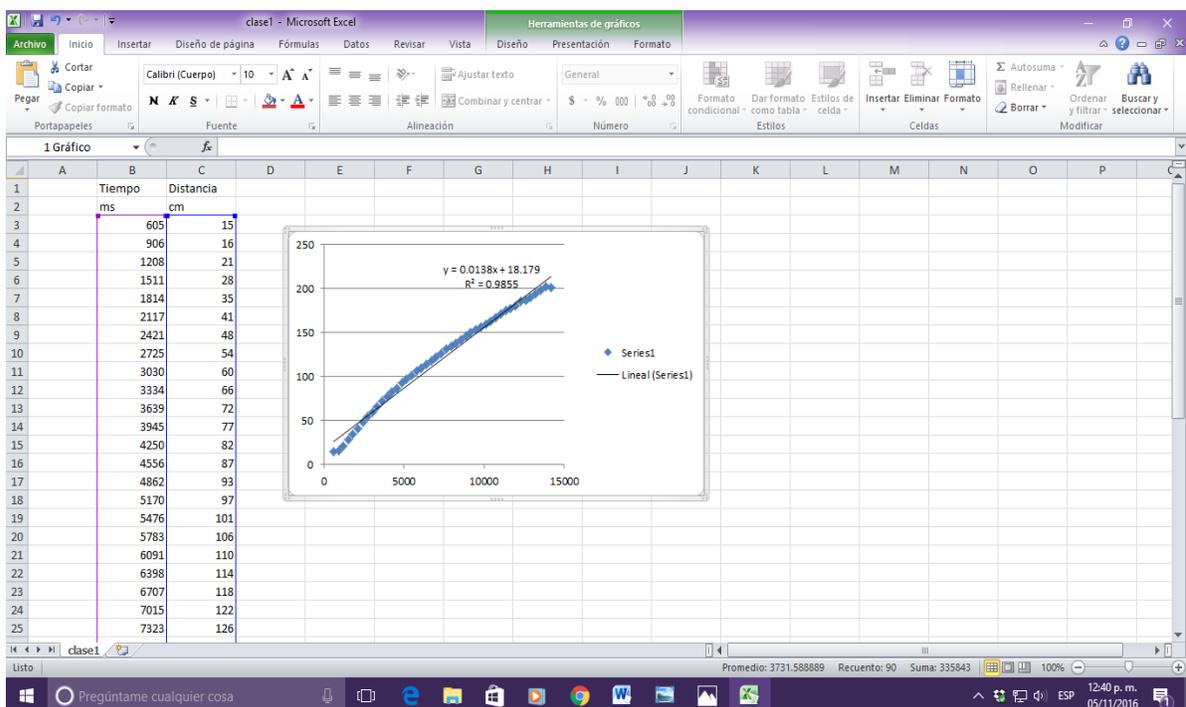


Figura 8. Datos y gráfica en una hoja de Excel para la primera fase.

2. Movimiento rectilíneo

Procediendo de la misma manera, para la fase dos o de movimiento rectilíneo uniforme acelerado, la Figura 9 muestra las tablas de datos y las gráficas, que se obtienen en esta fase. La primera gráfica, se obtuvo con los datos de tiempo vs distancia, la segunda con los datos del tiempo elevado al cuadrado vs distancia, y al igual que en la fase anterior, se dan las líneas de tendencia y las ecuaciones de éstas. Se puede ver en la primera gráfica, un comportamiento no lineal o cuadrático y en la segunda, un comportamiento lineal, donde la pendiente de esta gráfica representa la aceleración del sistema.

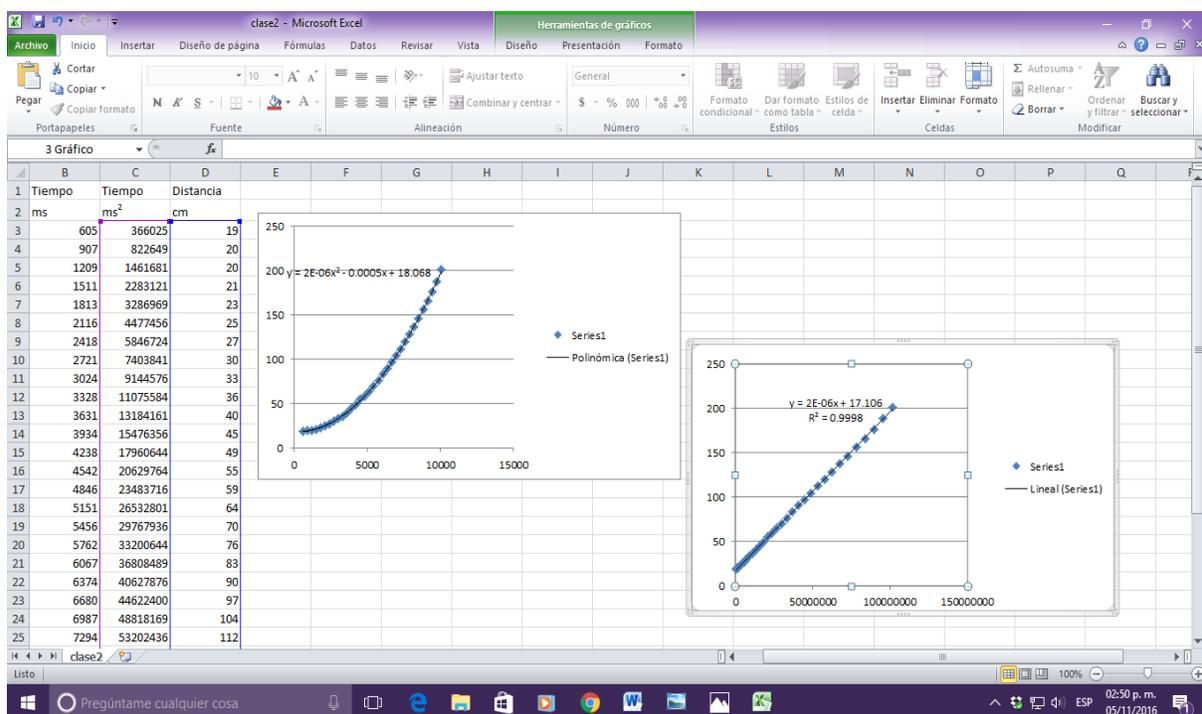


Figura 9. Datos y gráficas para la fase 2, movimiento rectilíneo uniforme acelerado.

En la fase dos se obtuvieron dos gráficas, la de posición contra el tiempo que representa una línea curva o parabólica, y la de posición contra el tiempo al cuadrado, la cual está representada por una línea recta. Para estas dos gráficas se pueden establecer algunas preguntas como las siguientes:

Para la primera gráfica:

- ¿Qué exponente tiene la ecuación que modela este comportamiento?,
- ¿Cuál es la ecuación que modela este comportamiento?,
- ¿Qué significado físico tiene la forma de la gráfica?,
- ¿Qué puede decirse de la velocidad del objeto?,
- ¿El objeto se aleja o se acerca al sensor?

Para la segunda gráfica:

- ¿Qué significado físico tiene la pendiente de la gráfica?
- ¿Cuáles son sus unidades?
- ¿Cuáles son sus unidades en el Sistema Internacional?,

2. Movimiento rectilíneo

- ¿Cuál es el modelo matemático que describe este comportamiento?,
- ¿La masa influye en este tipo de movimiento?

CONCLUSIONES

Las conclusiones a las que podrían llegar los alumnos podrían ser que:

- En un movimiento rectilíneo uniforme la velocidad permanece constante, ya que así lo indica la gráfica de posición contra tiempo, o que la velocidad no cambia.
- Un objeto moviéndose a velocidad constante y en línea recta continuará en ese estado de movimiento.
- La masa no influye en su movimiento.

En el movimiento rectilíneo uniforme acelerado las conclusiones de los alumnos podrían ser:

- La velocidad cambia en el tiempo.
- La aceleración permanece constante.
- El movimiento no depende de la masa.
- La ecuación que describe el movimiento depende del tiempo al cuadrado.

REFERENCIAS

Bueche, F. (1998). Fundamentos de Física. Quinta edición. México: Mc Graw Hill.

Giancoli, C. D. (2006). Física principios con aplicaciones. Sexta edición. México: Pearson Educación.

Tipler, A. P. (1995). Física. Tercera edición. España: Editorial Reverté, S.A.

Zitzewitz, P. W. Neff, R. y Davis, M. (2002). Física. Principios y problemas. México: Mc Graw Hill.

2. Movimiento rectilíneo

ANEXOS

1.- Programa "sensor de posición con tiempo" que se debe cargar en el Arduino para hacer la actividad experimental.

```
/*
Sensor de proximidad y al ser inferior a 10 cm
envía un pulso de alarma por el pin 13
HC-SR04 conexiones:
VCC al arduino 5v
GND al arduino GND
Echo al Arduino pin 11
Trig al Arduino pin 12
*/

#define echo 11
#define trig 12
long duracion, distancia;
unsigned long time = 0;           //declaración de tiempo

void setup() {
  Serial.begin (9600);           // inicializa el puerto seria a 9600 baudios
  pinMode(echo, INPUT);         // define el pin 11 como entrada (echo)
  pinMode(trig, OUTPUT);        // define el pin 12 como salida (trigger)
  pinMode(13, 1);               // Define el pin 13 como salida
  Serial.println (" Tiempo Distancia");
  Serial.println (" ms   cm ");
}

void loop() {
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);     // genera el pulso de trigger por 10 ms
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  duracion = pulseIn(echo, HIGH);
  distancia = (duracion/2) / 29; // calcula la distancia en centímetros

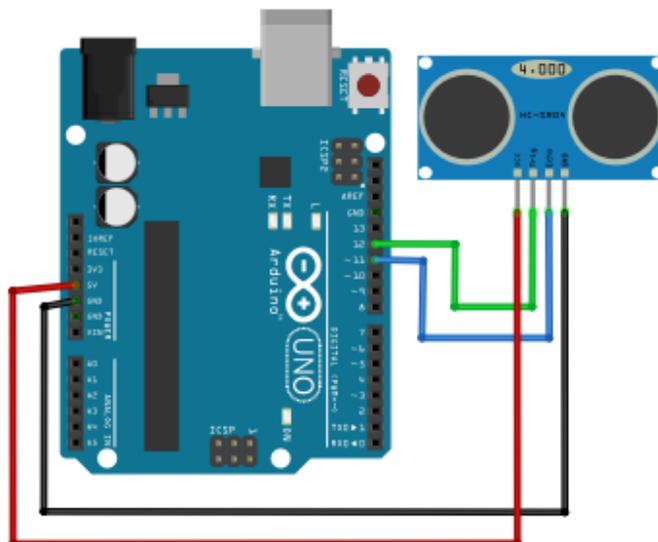
  //Serial.print(" Time: ");    // se dan espacios a la palabra "en esta posición
  Serial.print(" ");           //las instrucciones del tiempo
  time = millis();             //medir tiempo se imprimen primero"
  Serial.print (time);         //imprime el tiempo del Arduino

  if (distancia >= 500 || distancia <= 0){ // si la distancia es mayor a 500 cm o menor a 0 cm
    Serial.println("----"); // no mide nada
  }
  else {
    Serial.print(" "); // deja espacios antes del valor de la distancia
    Serial.println(distancia); // envía el valor de la distancia por el puerto serial
    digitalWrite(13, 0); // en bajo el pin 13
  }

  if (distancia <= 10 && distancia >= 1){
    digitalWrite(13, 1); // en alto el pin 13 si la distancia es menor a 10cm
    Serial.println("Alarma....."); // envía la palabra Alarma por el puerto serial
  }
  delay(300); // espera 300 ms para que se logre ver la distancia y el tiempo en la consola
}
```

2. Movimiento rectilíneo

2.- Diagrama esquemático del circuito electrónico que se conecta al Arduino.



3.- Otros

Adquisición de los datos con la ayuda del software.

Abrir el programa de Arduino sensor de posición con tiempo.

```
sensor_de_posici_n_con_tiempo4 Arduino 1.6.11
Archivo Editor Programa Herramientas Ayuda

sensor_de_posici_n_con_tiempo4

#define echo 11
#define trig 12
long duracion, distancia;

unsigned long time = 0; //declaracion de tiempo

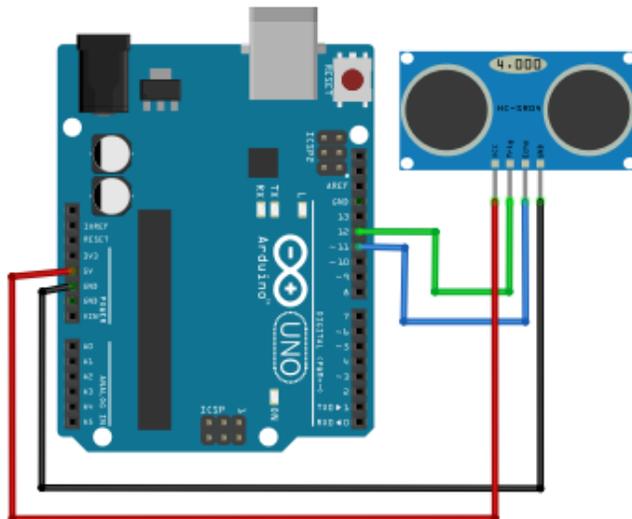
void setup() {
  Serial.begin (9600); // inicializa el puerto seria a 9600 baudios
  pinMode(echo, INPUT); // define el pin 11 como entrada (echo)
  pinMode(trig, OUTPUT); // define el pin 12 como salida (trigger)
  pinMode(13, 1); // Define el pin 13 como salida
  Serial.println (" Tiempo Distancia");
  Serial.println (" ms cm ");
}

Guardado.

El Sketch usa 2,800 bytes (8%) del espacio de almacenamiento de programa. El m
Las variables Globales usan 260 bytes (12%) de la memoria dinámica, dejando 1,
38 Arduino/Genuino Uno en COM4
```

2. Movimiento rectilíneo

Hacer las conexiones pertinentes de la tarjeta con el sensor.

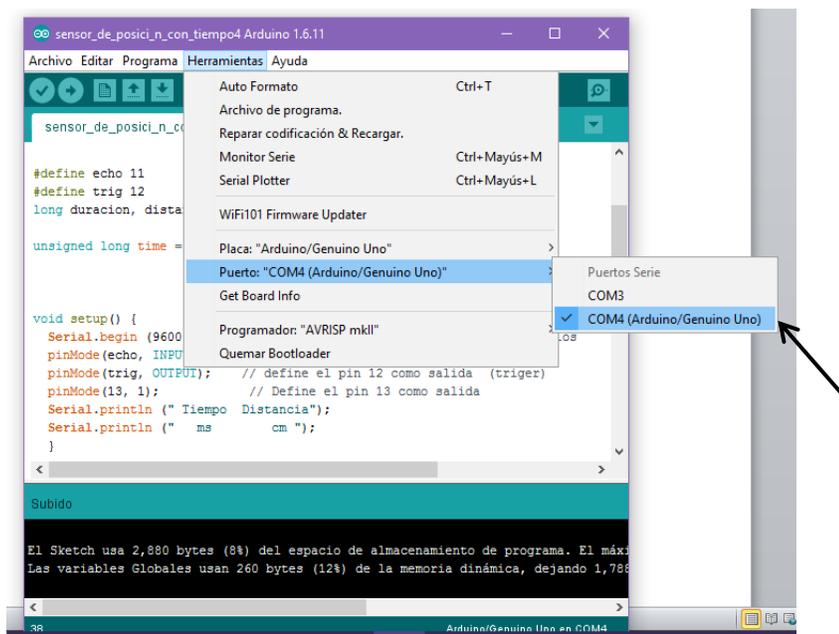


Conectar el sistema a la computadora.

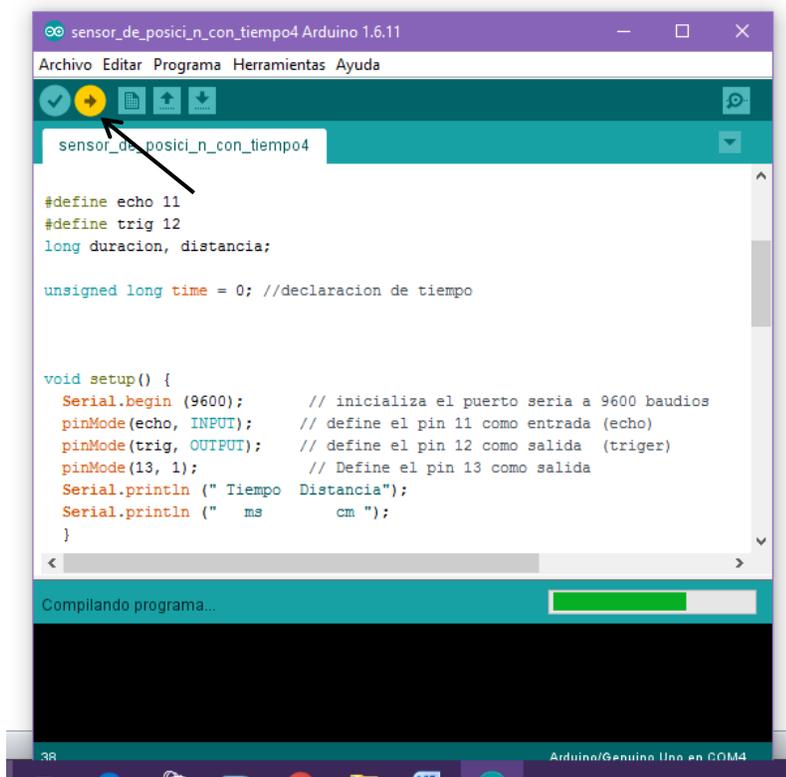


2. Movimiento rectilíneo

En la ficha herramientas del programa seleccionar el puerto de comunicación.

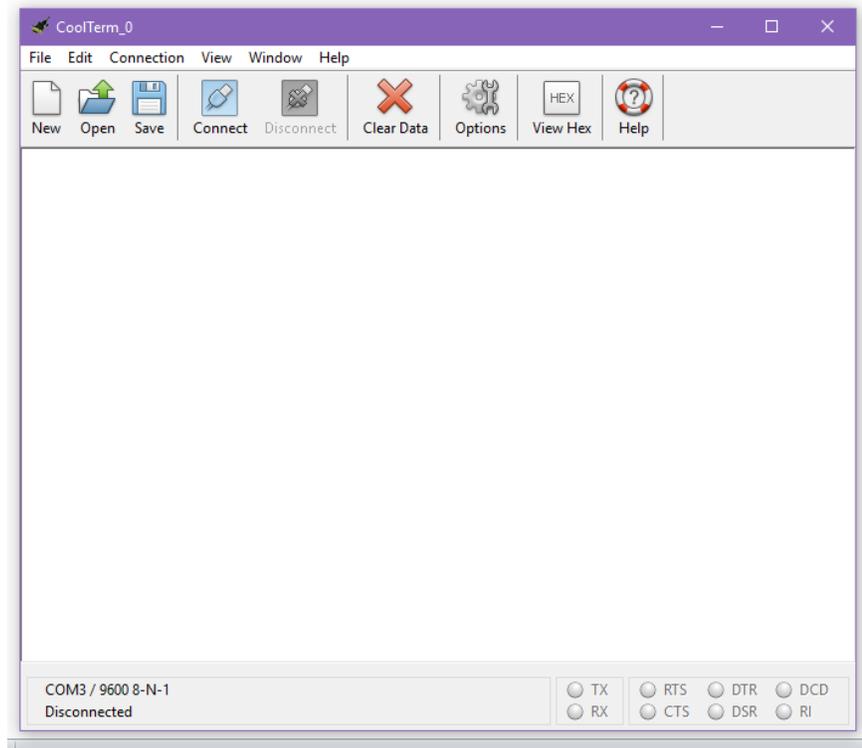


Hacer clic en la flecha para que se cargue el programa en el Arduino.

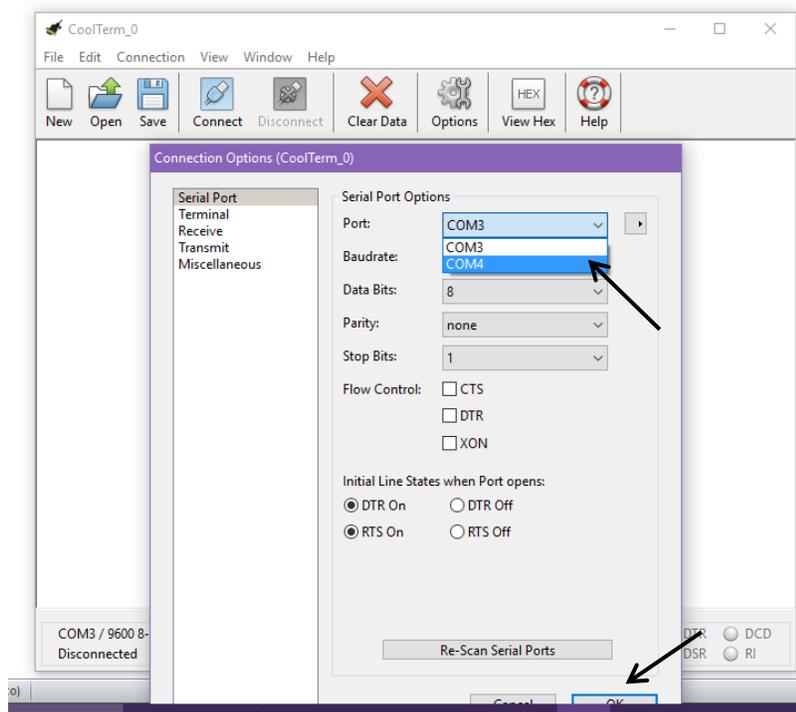


2. Movimiento rectilíneo

Abrir el programa Coolterm.

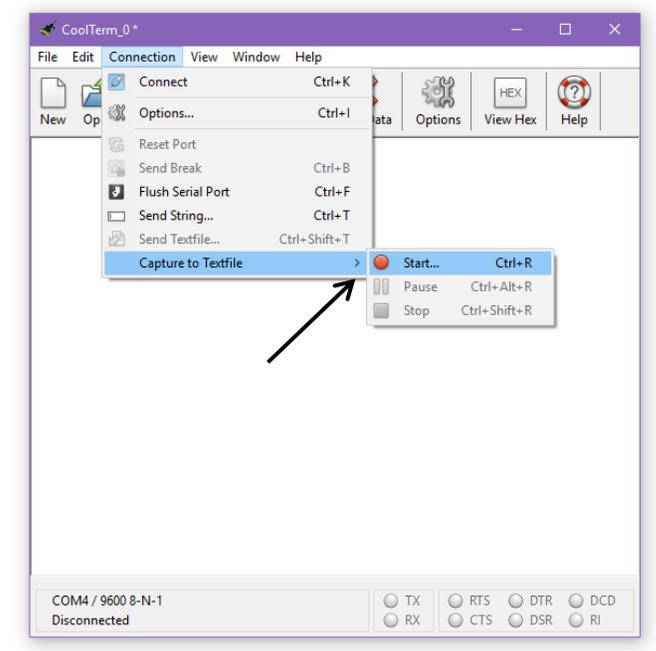


En opciones seleccionar el puerto que coincida con el Arduino y hacer clic en OK.

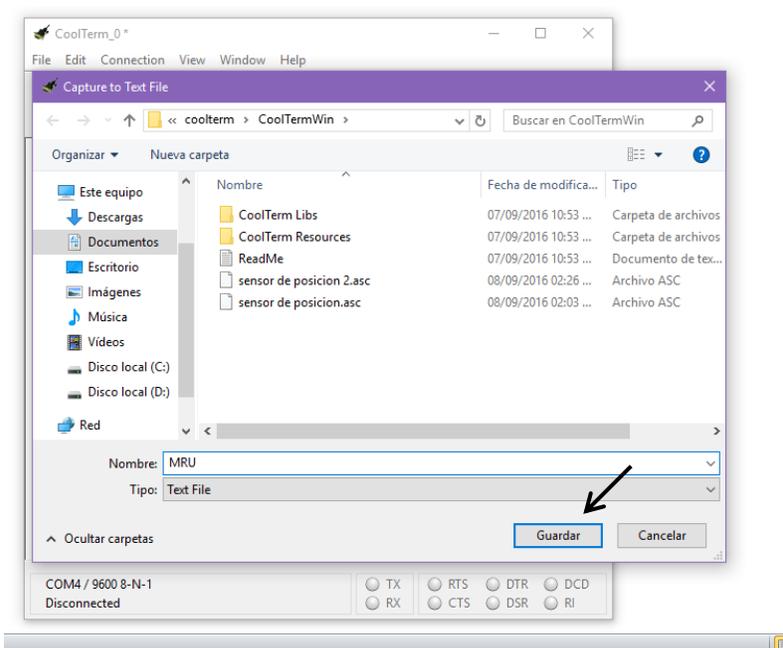


2. Movimiento rectilíneo

Seleccionar la pestaña Connection – Capture to Textfile – Start para que los datos se comiencen a guardar en un archivo.

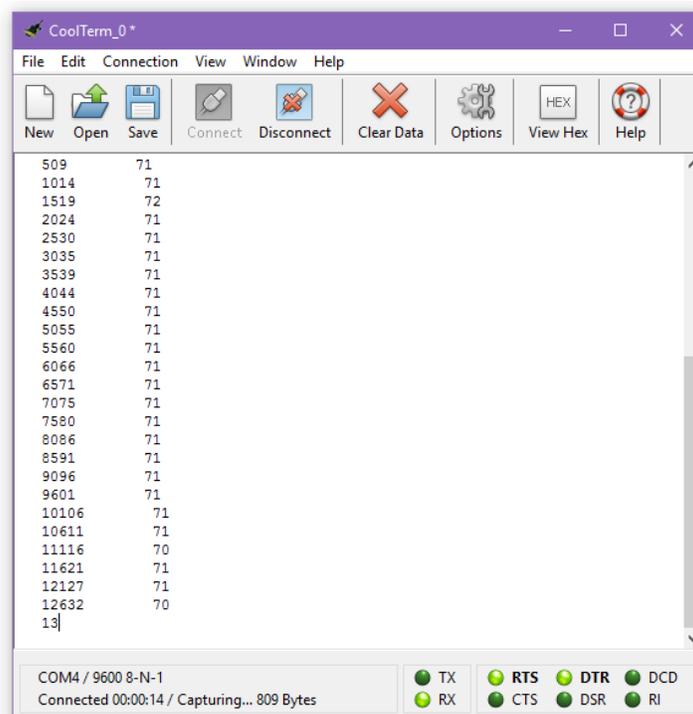


Aparecerá la siguiente ventana donde hay que seleccionar un nombre para el archivo y dar clic en guardar. Aquí es donde se van a guardar los datos adquiridos por lo que es importante conocer la carpeta y el nombre del archivo.

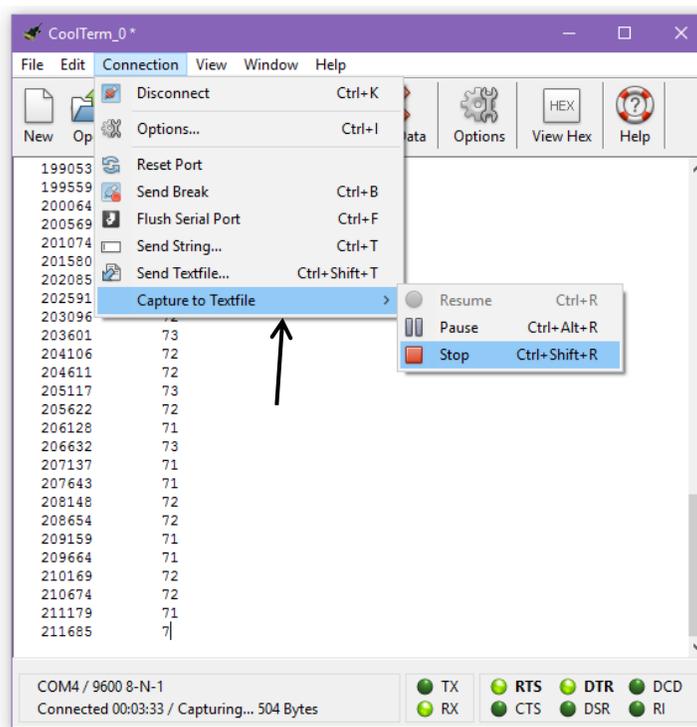


2. Movimiento rectilíneo

Dar clic en conectar y de manera casi inmediata aparecerán los datos.

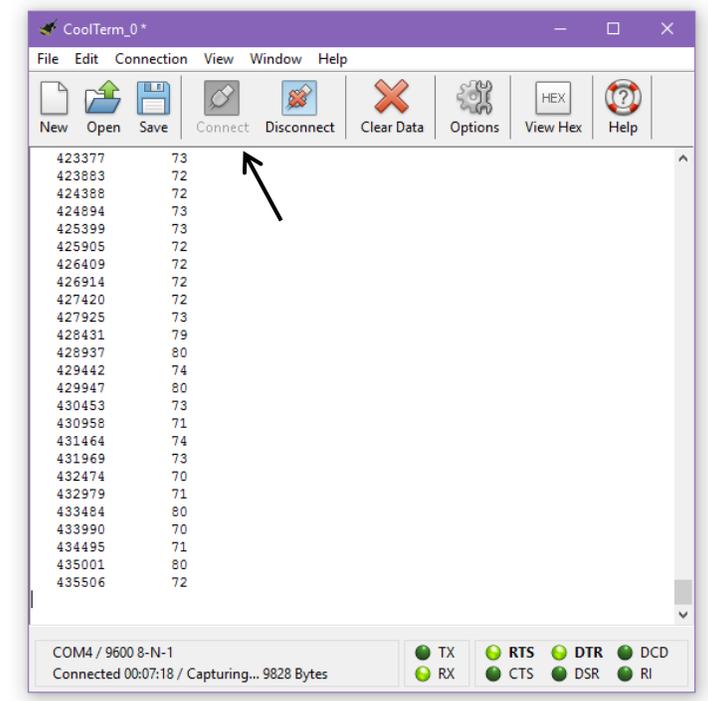


Hacer clic en la pestaña Connection – Capture to Textfile – Stop para que deje de capturar datos.

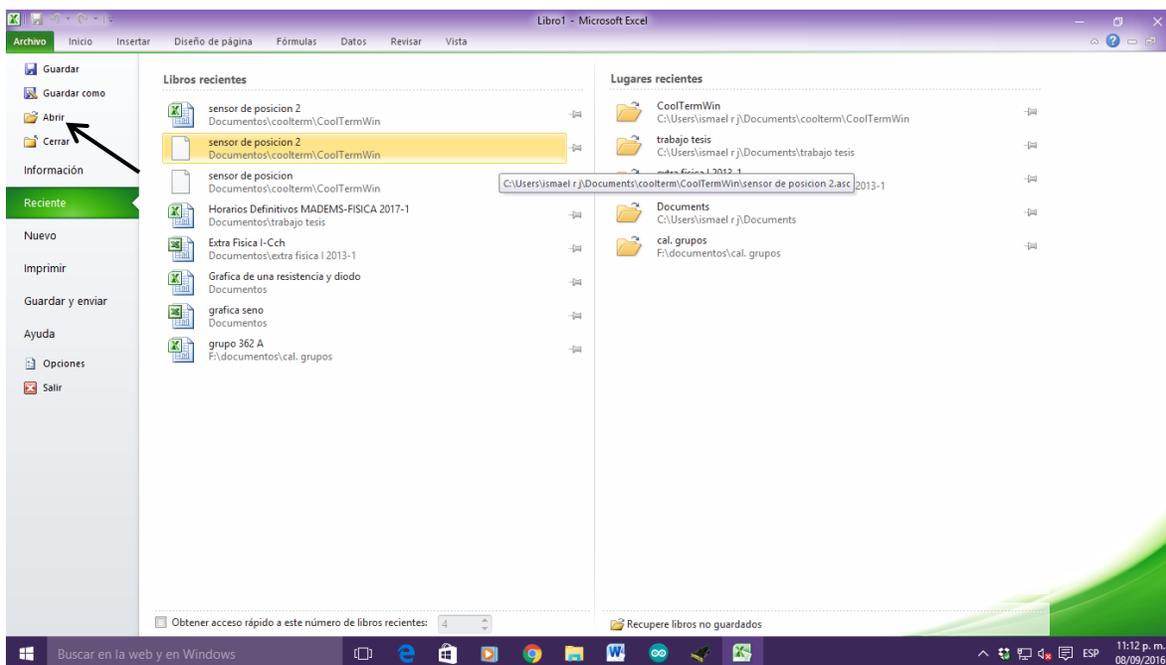


2. Movimiento rectilíneo

De manera casi inmediata hacer clic en el ícono Disconnect para que se pare completamente el sistema.

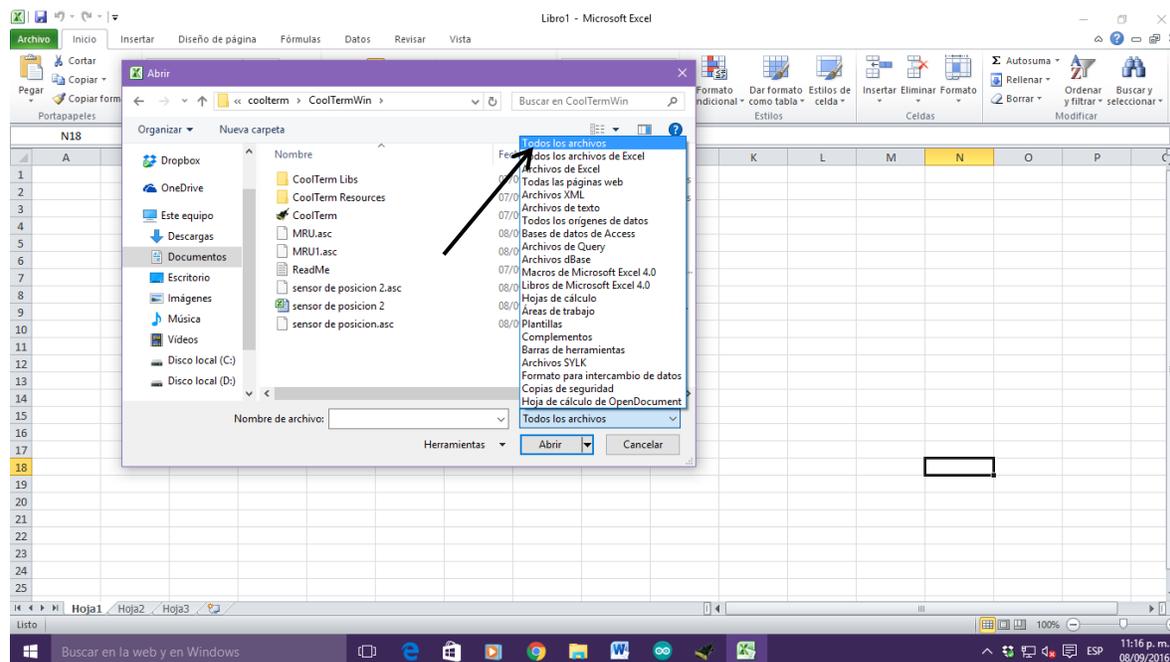


Abrir Excel hacer clic en la pestaña archivo – abrir.

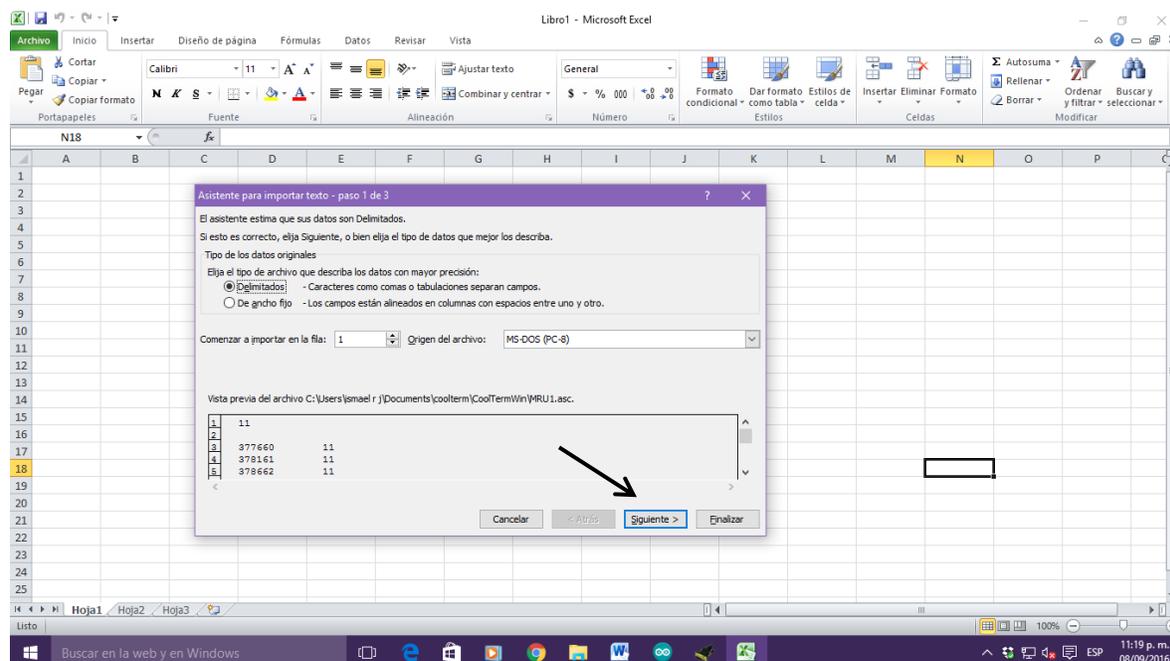


2. Movimiento rectilíneo

Buscar la carpeta que contiene el archivo de datos, seleccionar “todos los archivos” para que aparezca la carpeta.

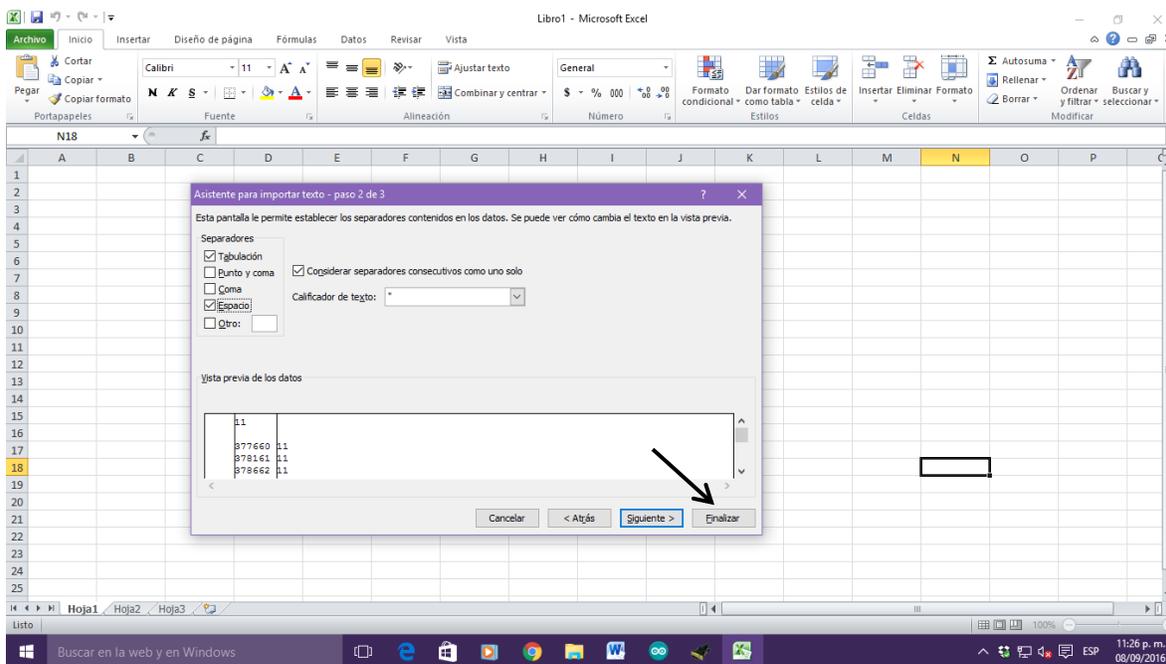


Seleccionar la carpeta, dar clic en abrir y aparece la siguiente ventana, dar clic en siguiente.

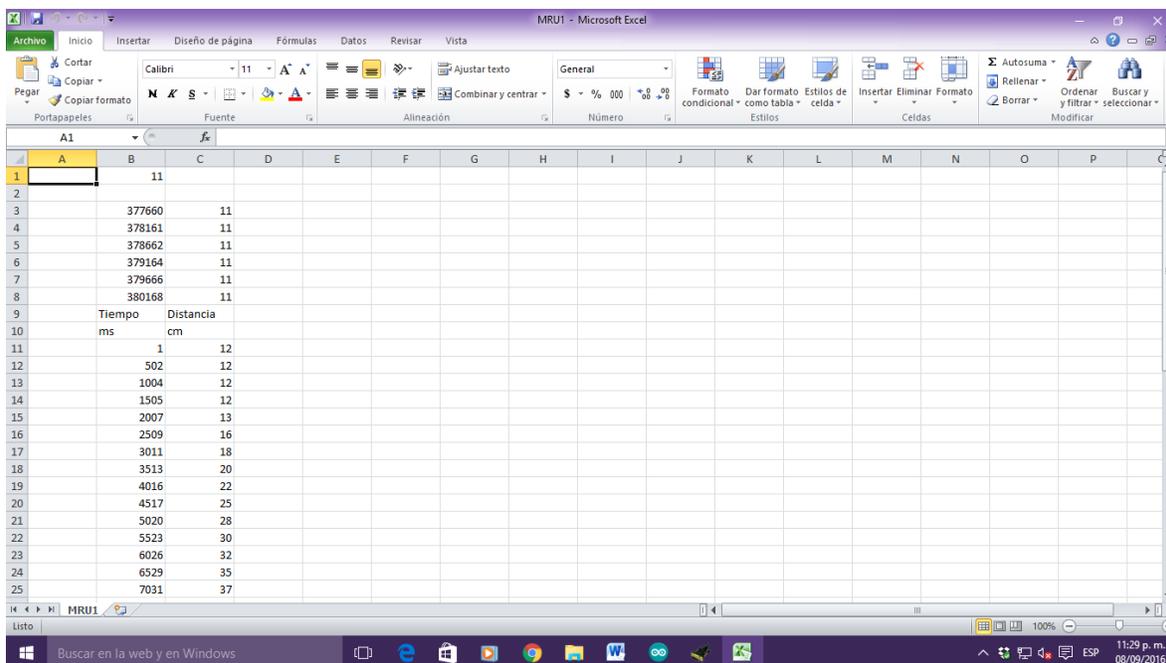


2. Movimiento rectilíneo

Aparecerá la siguiente ventana, seleccionar espacio y hacer clic en finalizar.

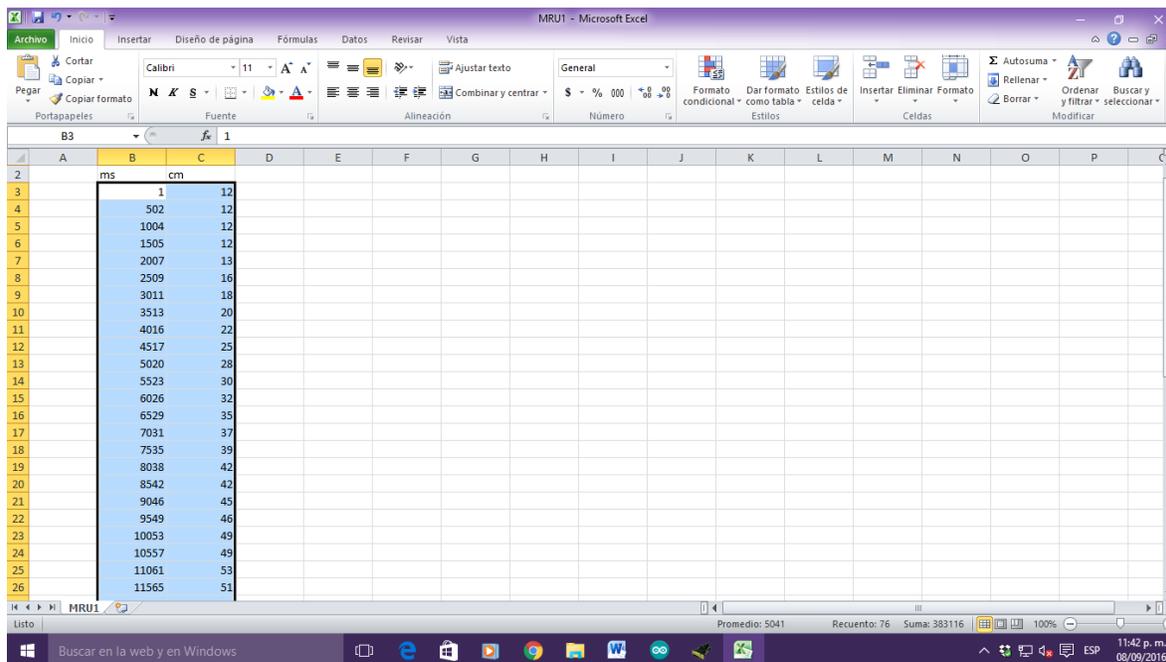


Aparecerá la ventana de Excel con los datos.

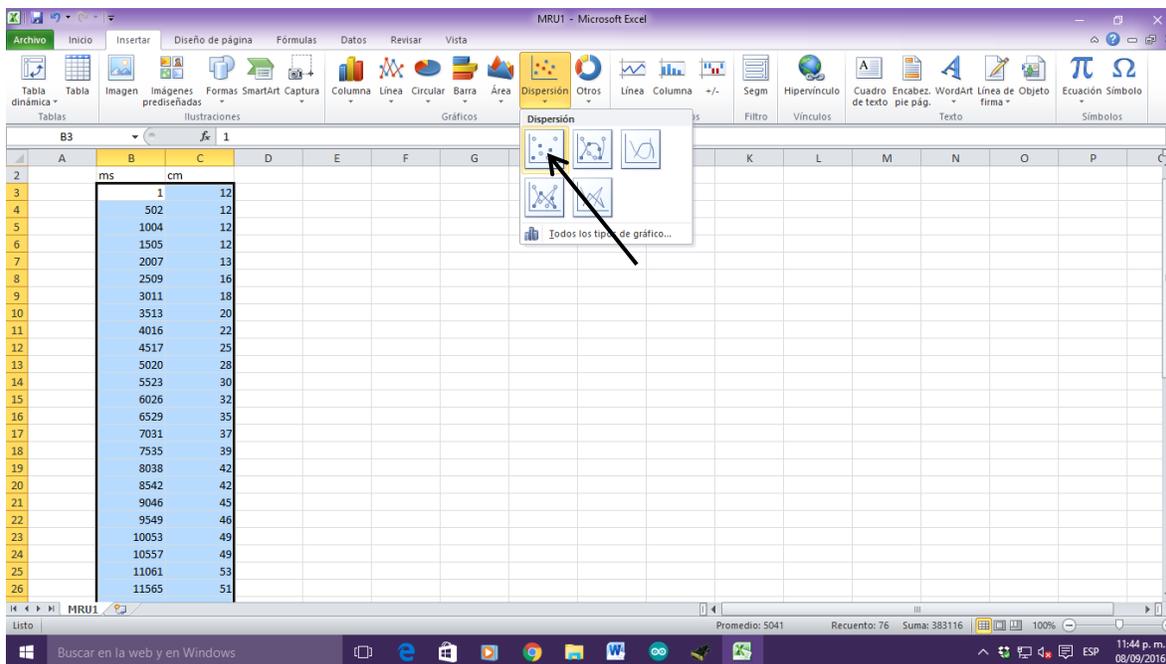


2. Movimiento rectilíneo

Seleccionar los datos para graficar.

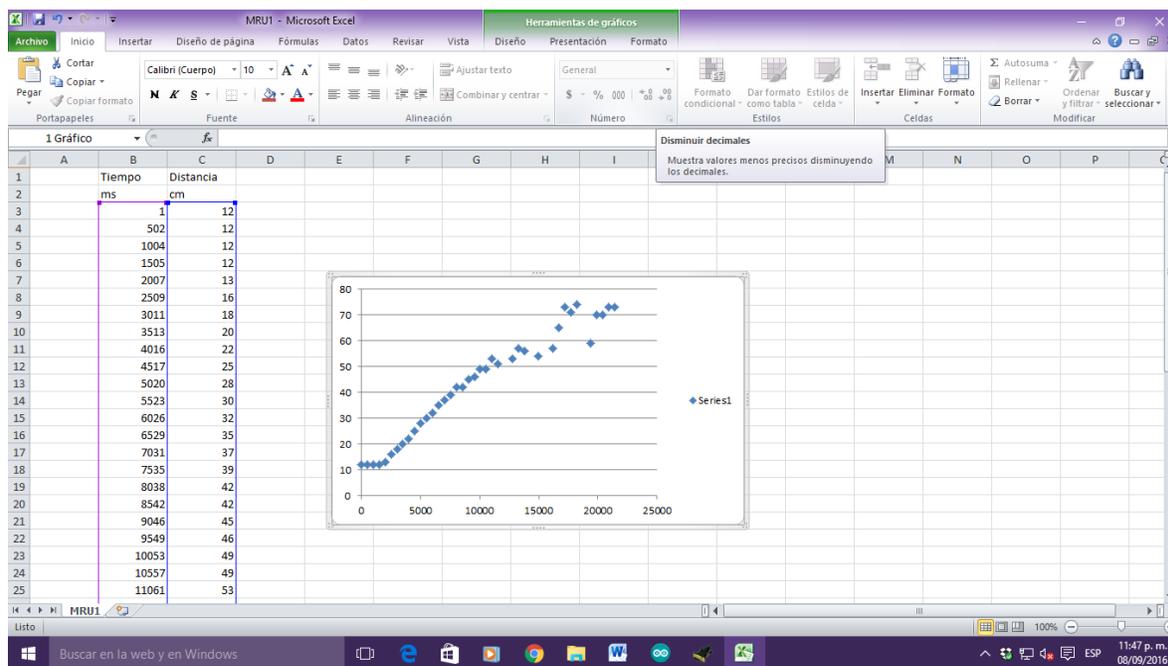


En la pestaña insertar, seleccionar dispersión y el tipo de gráfica.

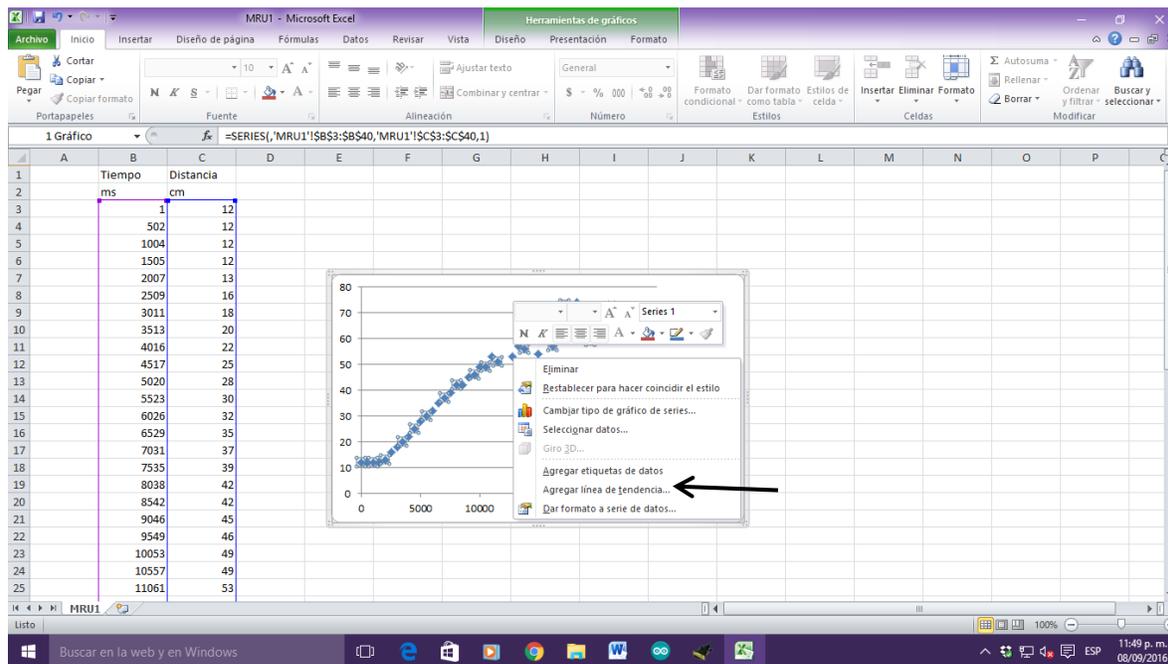


2. Movimiento rectilíneo

De manera inmediata aparece la gráfica.



Hacer clic derecho en alguno de los puntos para que aparezca el siguiente recuadro.



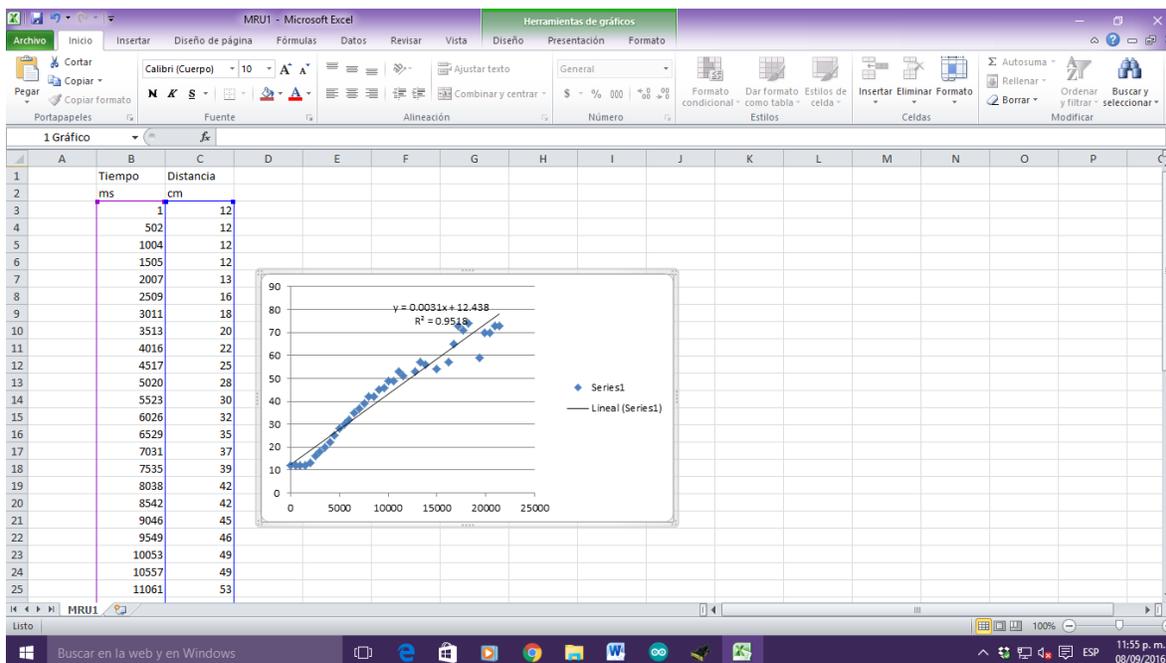
2. Movimiento rectilíneo

Seleccionar agregar línea de tendencia, activar la casilla de presentar ecuación en el gráfico y presentar el valor de R cuadrado en el gráfico, hacer clic en cerrar.

The screenshot shows the 'Formato de línea de tendencia' dialog box in Microsoft Excel. The 'Opciones de línea de tendencia' tab is selected. Under 'Tipo de tendencia o regresión', 'Lineal' is chosen. The 'Presentar el valor R cuadrado en el gráfico' checkbox is checked and highlighted by a black arrow. Other options like 'Exponencial', 'Logarítmica', 'Polinómica', 'Potencial', and 'Media móvil' are also visible. The 'Cerrar' button is at the bottom right.

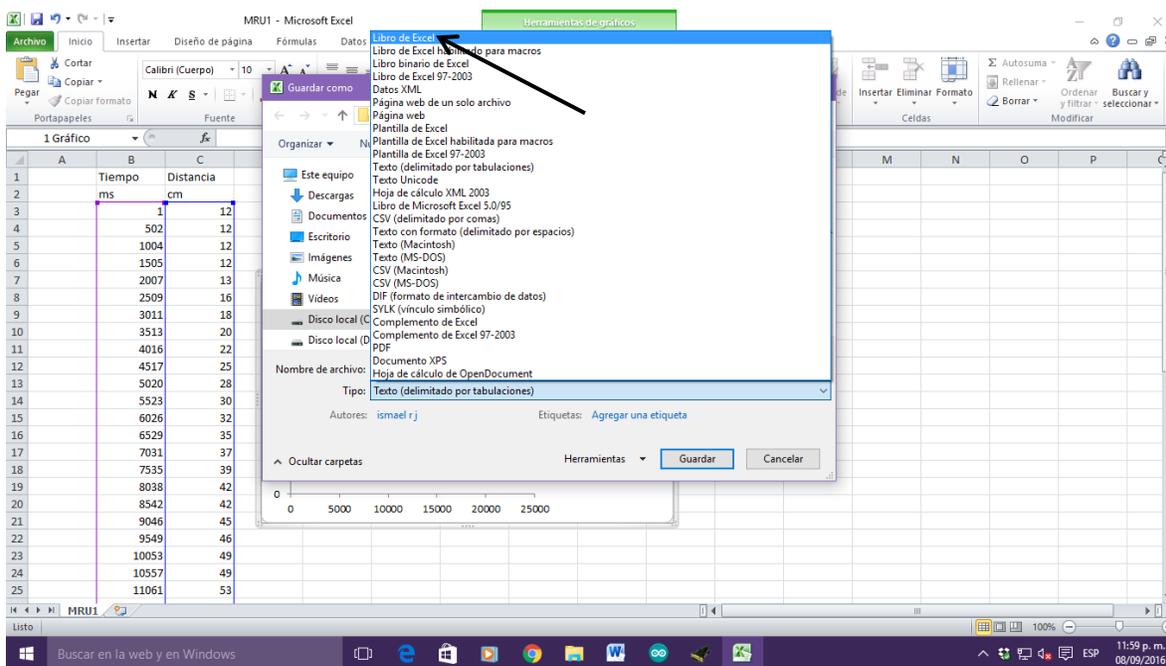
	A	B	C	D	E
1		Tiempo	Distancia		
2		ms	cm		
3		1	12		
4		502	12		
5		1004	12		
6		1505	12		
7		2007	13		
8		2509	16		
9		3011	18		
10		3513	20		
11		4016	22		
12		4517	25		
13		5020	28		
14		5523	30		
15		6026	32		
16		6529	35		
17		7031	37		
18		7535	39		
19		8038	42		
20		8542	42		
21		9046	45		
22		9549	46		
23		10053	49		
24		10557	49		
25		11061	53		

En la gráfica aparece la ecuación y el valor de R^2 .



2. Movimiento rectilíneo

Para guardar seleccionar archivo – guardar como – tipo – libro de excel y hacer clic en guardar.



3. MODELO “CASERO” DEL PROCESO DE RESPIRACIÓN

Responsable: Emilio García Valdez CCH-SUR	Participantes: Miguel Á. Bañuelos Saucedo ICAT	Revisión: Eduardo José Vega Murguía ICAT
---	--	--

Ubicación curricular

Colegio de Ciencias y Humanidades

Física I

Unidad 3. Energía: fenómenos térmicos, tecnología y sociedad

INTRODUCCIÓN

Al estudiar los gases experimentalmente, se pueden utilizar matraces o jeringas que encierran a un gas para cambiarle su presión, volumen o temperatura y observar las “leyes de los gases”. Estas actividades experimentales pueden ser de interés para varios alumnos, pero a muchos podría interesarles aplicar estas leyes a un fenómeno muy cercano a todos como lo es la respiración (ver figura 1).

GUÍA PARA EL PROFESOR: Para generar mayor interés de los alumnos en este fenómeno se sugiere plantear preguntas como las siguientes: ¿Ustedes respiran?, ¿cuántas respiraciones hacen por minuto?, ¿en una hora?, ¿en un día?

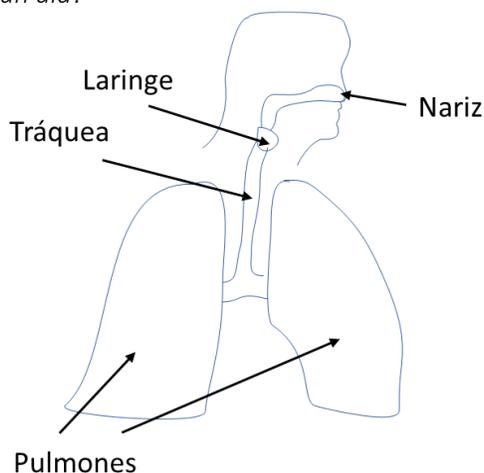


Figura 1. Esquema básico del aparato respiratorio humano.

De todas esas respiraciones que han realizado podrían explicar ¿cómo respiran? ¿Creen que el conocer las leyes de los gases puede ayudar a entender cómo respiran? Intenten explicar la respiración con las leyes de los gases.

Básicamente la respiración se lleva a cabo en el tórax y los pulmones del torso humano. El aparato respiratorio está compuesto básicamente de: fosas nasales, faringe, laringe tráquea, bronquios,

3. Modelo "casero" del proceso de respiración

pulmones y diafragma, que es un músculo y como todo músculo se contrae y relaja (Ver figuras 1 y 2.). El aire encerrado en el interior del tórax se puede considerar como un sistema cerrado en donde podemos aplicar las leyes de los gases.

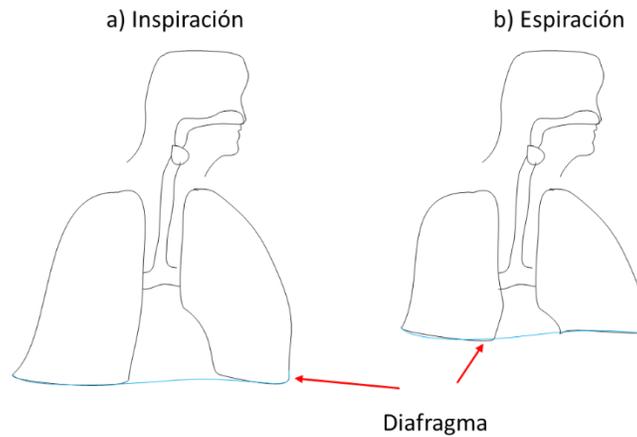


Figura 2. Papel del diafragma en el proceso de la respiración. a) Durante la inspiración se contraen los músculos pectorales menores y desciende el diafragma. b) Durante la espiración se relajan los músculos pectorales menores y el diafragma se eleva.

Es posible representar al tórax como un cilindro que encierra una cantidad de aire, a las fosas nasales como un orificio (y un tubito), que conecta al aire externo con un globo que simule al pulmón (dentro del cilindro-tórax). El diafragma puede simularse con otro globo que se coloca en la base del cilindro-tórax y que se pueda extender o contraer. De esta manera es posible construir un simulador casero de la respiración (ver figura 3).

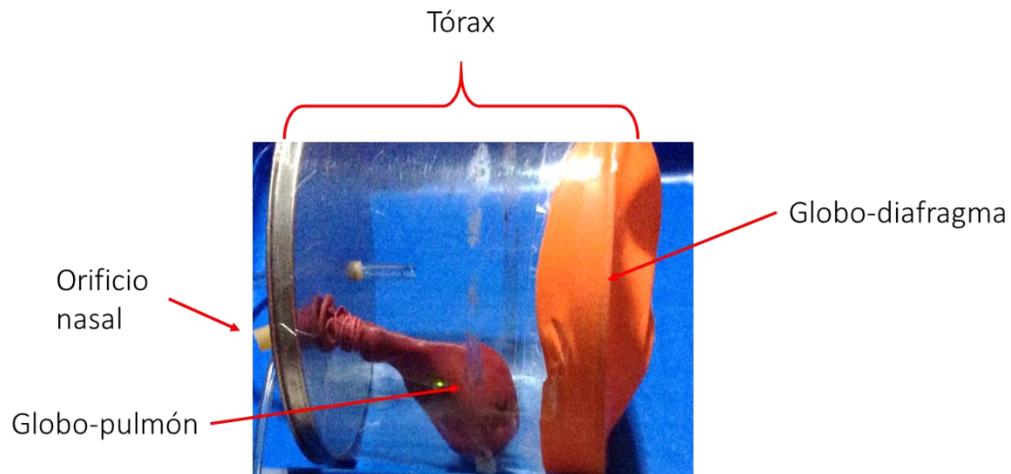
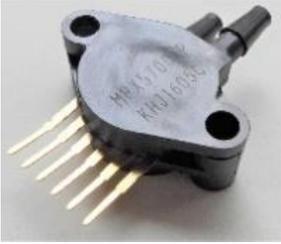


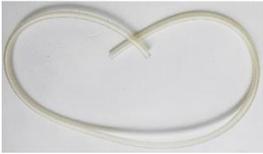
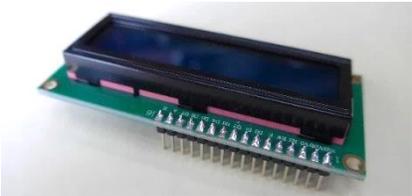
Figura 3. Modelo ensamblado de tórax.

3. Modelo “casero” del proceso de respiración

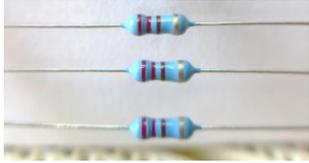
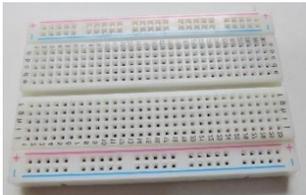
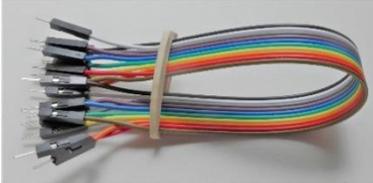
MATERIAL

Descripción	Imagen	Comentarios
Tarjeta Arduino UNO		
Cable USB A macho a B macho		Conecta la tarjeta Arduino con la Laptop.
Sensor de presión		Sensor MPX5700DP para manómetro. Compara la presión con la de la atmósfera. El pin con una pequeña muesca es el pin número 1. Sólo se conectan los pines 1, 2 y 3.
Bote de plástico		Simulará el Tórax
Cuchillo o tijeras		
3 globos, 2 pequeños y 1 grande		

3. Modelo "casero" del proceso de respiración

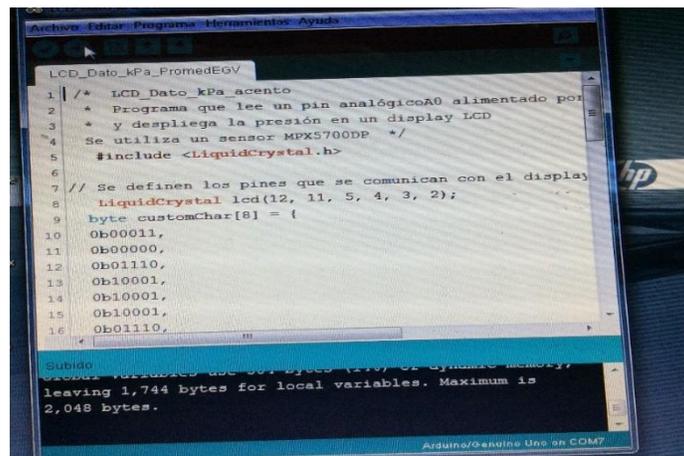
Descripción	Imagen	Comentarios
1 m de manguera de plástico		Puede utilizarse manguera para acuario.
Clavo grande y martillo		
Barra de silicón y pistola		
Ligas y cinta adhesiva		
Vaso de precipitado o cinta métrica		
Pantalla de cristal líquido de 16 x 2 caracteres		Con pines soldados por el usuario.

3. Modelo "casero" del proceso de respiración

Descripción	Imagen	Comentarios
Tres resistencias		Carbón, ½ Watt. 220 ohms 1 kohm 8.2 kohm
Protoboard conexiones. 400		
Cables de conexión. Jumpers macho-macho 20 cm		

Software

- Programa Arduino IDE 1.8.1 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>.
- Programa CoolTerm 1.4.7 o superior. Disponible en: <http://freeware.the-meiers.org/>.
- Programa para el modelo de pulmón. Proporcionado por el profesor (ver figura 4).



```

LCD_Dato_kPa_PromedEsv
1 | /* LCD_Dato_kPa_acento
2 | * Programa que lee un pin analógicoA0 alimentado por
3 | * y despliega la presión en un display LCD
4 | * Se utiliza un sensor MPX5700DP */
5 | #include <LiquidCrystal.h>
6 |
7 | // Se definen los pines que se comunican con el display
8 | LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
9 | byte customChar[8] = {
10 | 0b00011,
11 | 0b00000,
12 | 0b01110,
13 | 0b10001,
14 | 0b10001,
15 | 0b10001,
16 | 0b01110,
17 | };
18 |
19 | void setup() {
20 |   lcd.begin(16, 2);
21 |   lcd.setCursor(0, 0);
22 |   lcd.print("Presión");
23 | }
24 |
25 | void loop() {
26 |   float p = analogRead(A0);
27 |   p = p * 0.0001;
28 |   lcd.setCursor(0, 1);
29 |   lcd.print(p);
30 |   delay(1000);
31 | }
32 |
33 | Subido
34 | Global variables use 60 bytes (11%) of dynamic memory,
35 | leaving 1,744 bytes for local variables. Maximum is
36 | 2,048 bytes.
37 |
38 | Arduino/Genuino Uno on COM7
  
```

Figura 4. Ventana principal del sistema Arduino mostrando el programa.

3. Modelo “casero” del proceso de respiración

DESARROLLO

1.- CONSTRUCCIÓN DEL MODELO. Hacer dos orificios (con el clavo) en tapa del bote de plástico (que constituirá el cilindro-tórax), posteriormente hacer varios orificios en la base hasta lograr introducir un cuchillo o unas tijeras para poder separar toda la base. De esta manera se obtendrá un cilindro con una tapa de un lado y una boca abierta del otro. En cada orificio de la tapa se debe introducir un pedazo de manguera de aproximadamente 4 cm de largo y pegarlo con silicón. Una manguera se conectará a un globo que funcionará como pulmón y se apretará la conexión mediante ligas. La segunda manguera se conectará al sensor de presión que funciona con el Arduino. En seguida se deberá cortar otro globo (se requiere que sea un globo grande) y pegarlo o adherirlo con ligas o cinta adhesiva a la base del cilindro (para simular el diafragma). Es importante revisar que no haya fugas del aire interior.

2- Se jala hacia afuera el globo-diafragma para observar lo que ocurre y cerciorarse que el aire exterior sólo entre por el orificio de la tapa y que no haya fugas.

3.- Se enciende el medidor de presión del Arduino y se vuelve a jalar el diafragma-globo para medir la presión del aire interior del cilindro-tórax. Se presiona hacia adentro el globo-diafragma para medir la nueva presión.

4.- Se comparan los resultados con la ley de Boyle-Mariotte para gases, que establece que: para una masa dada de gas a temperatura constante, si aumentamos su volumen la presión disminuirá y viceversa.

El montaje completo se muestra en la figura 5. Los diagramas de conexión del sensor de presión y de la tarjeta Arduino se incluyen en el Anexo.

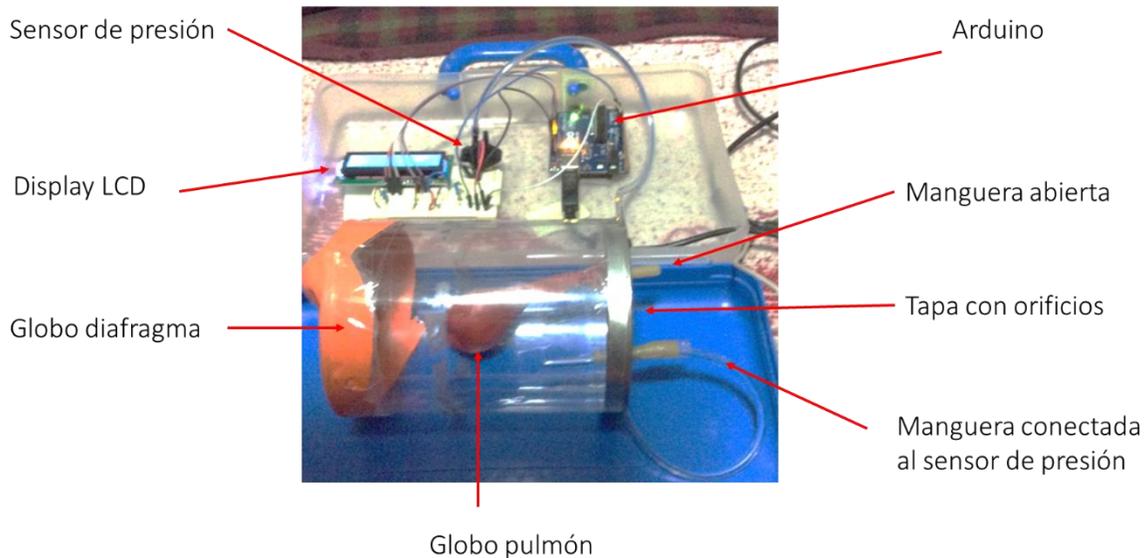


Figura 5. Detalle del modelo conectado al sistema Arduino.

3. Modelo “casero” del proceso de respiración

Opcionalmente, se puede conectar el sistema a una computadora personal y obtener los valores de la presión mediante la función “monitor serial” del ambiente Arduino IDE, como se muestra en la figura 6. El programa muestra simultáneamente el resultado en la pantalla LCD, como se ilustra en la figura 7.



Figura 6. Conexión del sistema a una computadora portátil.

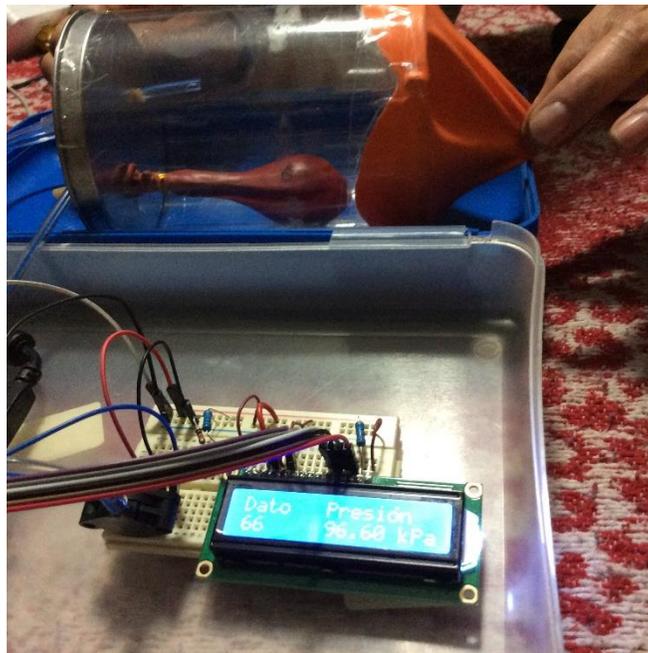


Figura 7. Detalle del display LCD indicando la presión.

RESULTADOS

Presión del cilindro en reposo: $P_1 = 91.49$ KPa.

Presión del cilindro con el diafragma extendido: $P_2 = 89.73$ KPa.

3. Modelo “casero” del proceso de respiración

CÁLCULOS

Volumen y presión del cilindro-tórax en reposo:

$$\pi = 3.1416 \quad R = 5 \text{ cm} \quad h_1 = 14.2 \text{ cm} \quad y \quad P_1 = 91.49 \text{ kPa}$$

Donde R es el radio del bote de plástico y h_1 su altura con el diagrama en reposo.

$$V_1 = \pi R^2 h_1 = 3.1416 (5\text{cm})^2(14.2\text{cm}) = 1115.268 \text{ cm}^3 = 1.115 \text{ L}$$

Volumen y presión del cilindro jalando el globo-diafragma:

$$P_2 = 89.73 \text{ kPa} \quad y \quad h_2 = 15 \text{ cm}$$

$$V_2 = \pi R^2 h_2, V_2 = V_1 + \text{Volumen del cono del diafragma}$$

$$V_2 = 1115.268 \text{ cm}^3 + 3.1416 (5 \text{ cm})^2 ((15-14.2)/3) = 1135.93 \text{ cm}^3 = 1.1359 \text{ L}$$

Si multiplicamos la presión por el volumen en los dos casos obtendremos:

$$P_1 V_1 = 102.035 \text{ L kPa} \quad y \quad P_2 V_2 = 101.92 \text{ L kPa}$$

Estos resultados concuerdan aproximadamente con la ley de Boyle-Mariotte para gases ligeros (ver figura 8).

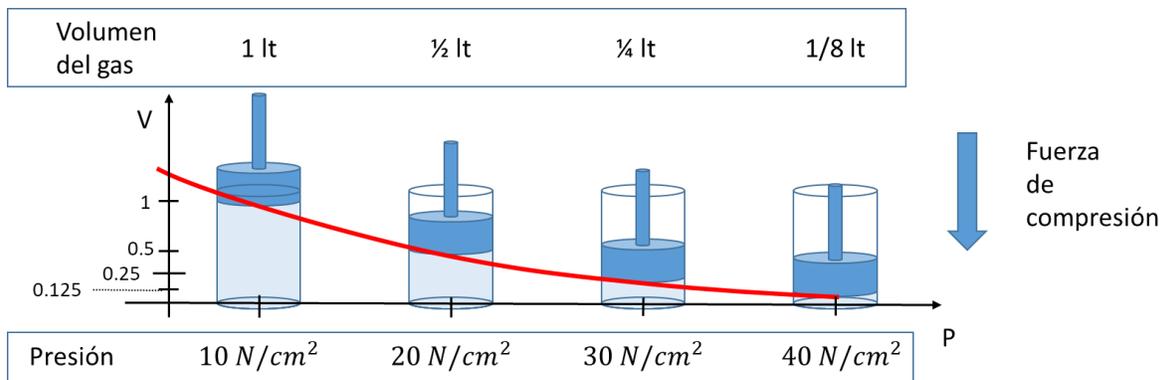


Figura 8. Representación de la ley de Boyle-Mariotte.

Preguntas para reflexionar, analizar y ayudar a las conclusiones:

- ¿Cómo cambia la presión dentro del bote comparada con la presión externa, al jalar el globo diafragma?
- ¿Qué cambio observan en el globo-pulmón? ¿Por qué cambió así el globo?
- ¿Cómo es la presión del aire que está en la atmósfera del exterior, respecto a la presión del aire que rodea al globo-pulmón, cuando se jala el globo-diafragma?
- ¿Qué le ocurre al globo-pulmón cuando se deja de jalar el globo-diafragma de la base? ¿Por qué ocurre?
- ¿Cómo se aplica la ley de Boyle – Mariotte para explicar lo que ocurre en este modelo de respiración?
- ¿Cómo explicarían ahora lo que les ocurre a los pulmones cuando se inhala y se exhala?

Lectura complementaria

3. Modelo “casero” del proceso de respiración

Se puede hacer una comparación con lo reportado en la literatura sobre el tema y observar que sí hay coincidencias, aunque las medidas son aproximadas. En la inhalación, el diafragma se contrae y el diafragma baja y la cavidad torácica se amplía. Esta contracción crea un vacío que succiona el aire hacia los pulmones. En la exhalación, el diafragma se relaja y el aire es expulsado de los pulmones. El aire entra en los pulmones porque la presión interna de estos es inferior a la atmosférica y por lo tanto existe un gradiente de presión. Inversamente, el aire es expulsado de los pulmones cuando estos ejercen, sobre el aire contenido, una presión superior a la atmosférica. Al subir el globo (diafragma) se produce una disminución de volumen en los pulmones (globo), por lo que la presión de aire interno (sistema) aumenta, entonces la mayor presión de aire hace que el aire salga de los pulmones a través de la garganta hacia el exterior, porque la presión del aire exterior es mayor a la del aire interior.

DISCUSIÓN Y CONCLUSIONES

La conclusión principal es que gracias a la ley de Boyle – Mariotte es posible explicar el proceso de inhalación y exhalación de la respiración humana. Aunque no podemos asegurar que el proceso sea realmente a temperatura constante y que el aire sea un gas ligero. Pero se puede considerar como una primera aproximación al entendimiento del proceso de la respiración humana, el cual es más completo y complejo.

REFERENCIAS

- Cromer, A. (1981). Física para las ciencias de la vida. Segunda edición. España: Editorial Reverté.
- Timberlake, K. (1997). Química. Introducción a la Química General, a la Orgánica y a la Bioquímica. Quinta edición. México: Oxford University Press – Harla.

3. Modelo "casero" del proceso de respiración

ANEXOS

1.- Programa que se debe grabar en Arduino para hacer el experimento

```
/* * Programa que lee un pin analógico A0 alimentado por un sensor de presión
 * y despliega la presión en un display LCD
 Se utiliza un sensor MPX5700DP */
#include <LiquidCrystal.h>

// Se definen los pines que se comunican con el display
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
byte customChar[8] = { //Genera un carácter "ó" acentuado
0b00011,
0b00000,
0b01110,
0b10001,
0b10001,
0b10001,
0b01110,
0b00000
};
const int pinSensor = 0; // A0 es el pin del sensor analógico de presión
const float ajusteCero = -46.2; // Hace el ajuste a cero del sensor
int valorSensor = 0; // Guarda el valor del sensor
float presion, voltaje;
// Inicialización
void setup() {

    Serial.begin(9600); //Configura la comunicación serial

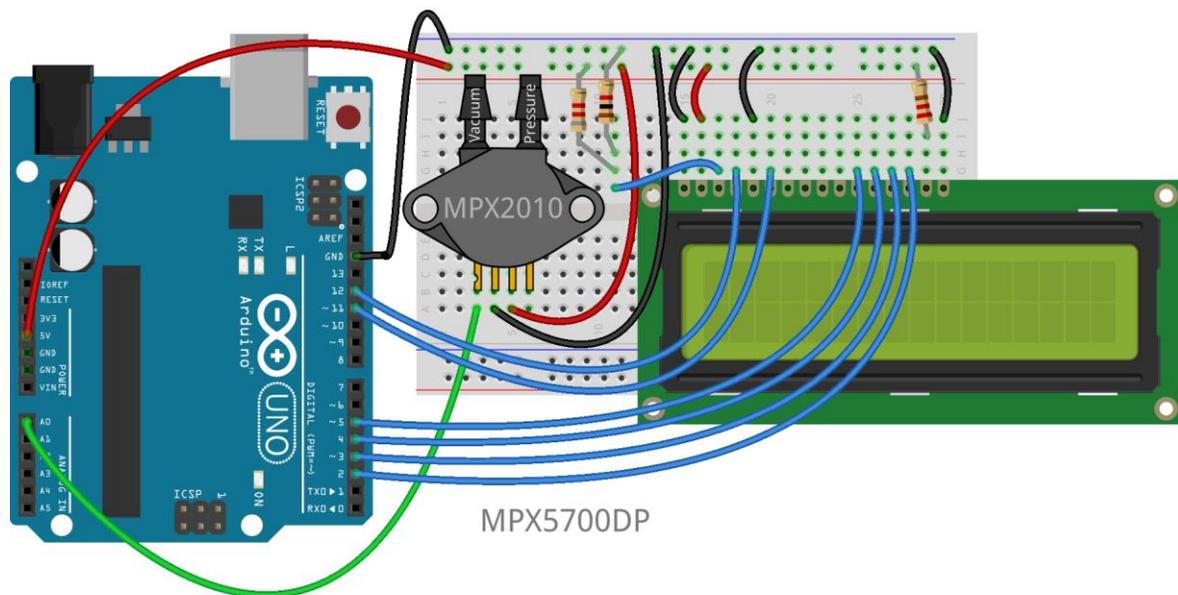
    lcd.createChar(0, customChar); // Crea el carácter "ó"
    lcd.begin(16, 2); // LCD es de 16 caracteres y 2 renglones

// Manda un letrero al display
    pinMode(pinSensor, INPUT);
    lcd.print("Dato Presi");
    lcd.write((byte)0);
    lcd.print("\n");
}
// Programa principal
void loop() {
    // lcd.setCursor(col, row)
    valorSensor = analogRead(pinSensor);
    lcd.setCursor(0, 1); // Selecciona caracter 0, fila 1 (segunda)
    lcd.print(" "); // borra 16 caracteres
    lcd.setCursor(0,1);
    lcd.print(valorSensor);
    voltaje = 5000*(valorSensor/1023.0); // Convierte a mV el valor de la presión
    presion = voltaje/6.4 - ajusteCero; // Convierte a kPa
```

3. Modelo "casero" del proceso de respiración

```
lcd.setCursor(7, 1);  
lcd.print(presion);  
lcd.print(" kPa");  
delay(450);  
Serial.print("La presion es de: "); // Envía el resultado por el puerto serial  
Serial.println(presion );  
}
```

2.- Diagrama esquemático del circuito electrónico que se conecta al Arduino



4. EQUILIBRIO TÉRMICO

Responsable: Gonzalo Víctor Rojas Cárdenas CCH Oriente	Participantes: Miguel Á. Bañuelos Saucedo ICAT	Revisión: Eduardo José Vega Murguía ICAT
--	--	--

Ubicación curricular

Colegio de Ciencias y Humanidades

Física I

Unidad 3. Energía: fenómenos térmicos, tecnología y sociedad

INTRODUCCIÓN

Por experiencia cotidiana sabemos que una bebida caliente, como el té o la leche, después de cierto tiempo de ser dejada en una taza se enfría hasta alcanzar la temperatura ambiente. Este hecho lo ha presenciado todo el mundo a lo largo de su vida, un sinnúmero de veces. Como conocimiento empírico ha permanecido por muchas generaciones, sin embargo, de una forma particular, Leonardo da Vinci, Galileo y otros, sabían que al contacto con un tercer cuerpo, usualmente el aire, dos o más cuerpos en contacto con él, se “mezclaban” de una manera apropiada, hasta alcanzar una misma “condición”. Así también, Joseph Black dice: – podemos percibir la tendencia del calor a difundirse de cualquier cuerpo caliente hacia otros más fríos en sus alrededores, hasta que el calor se distribuye entre ellos de una manera tal que ninguno es capaz de tomar más que los restantes. Así que, el calor alcanza un estado de equilibrio – (García-Colín, 2008).

A esta fenomenología la conocemos como equilibrio térmico. Actualmente se expresa como el principio de equilibrio, el cual dice que siempre que algunos objetos se coloquen juntos en un ambiente *aislado*, finalmente alcanzarán la misma temperatura, como resultado de la transferencia de energía térmica de los cuerpos más calientes a los cuerpos más fríos. Por consecuencia, el calor perdido por los cuerpos calientes debe ser igual al calor ganado por los cuerpos fríos. Es decir,

$$Q_{perdido} = Q_{ganado} \quad (1)$$

Esta ecuación expresa el resultado neto de la transferencia de calor dentro del sistema (Tippens, 2011).

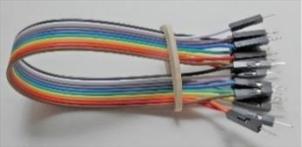
En esta práctica se propone una forma de estudiar el comportamiento térmico de un sistema líquido, que evoluciona hacia el equilibrio térmico, para verificar el cumplimiento de este principio. El instrumento de medida consta de dos termómetros digitales DS18B20 acoplados a una placa Arduino UNO. Se presenta también, el análisis resultante de los datos medidos.

4. Equilibrio térmico

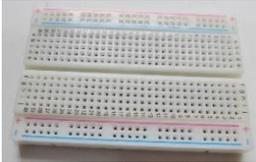
OBJETIVOS

El alumno comprenderá el fenómeno de equilibrio térmico como la consecuencia de transferencia de energía entre sistemas, debido a diferencias de temperatura.

MATERIAL

Descripción	Imagen	Comentarios
Tarjeta Arduino UNO		
Cable USB A macho a B macho		
Dos sensores de temperatura DS18B20		Modelo para inmersión en líquido.
Dos vasos de Unigel de 1 litro de capacidad, con tapa		En México al polietileno espumado se le conoce como: Unigel.
Quince cables tipo jumper macho-macho		

4. Equilibrio térmico

Descripción	Imagen	Comentarios
Un vaso pequeño de plástico, aproximadamente de 200 mL		Procurar que el grosor, de la pared del vaso, sea muy delgada. Se puede utilizar un vaso de los de Yogurt.
Dos resistencias de 4.7 k Ω , una resistencia de 8.2 k Ω , una de 1 k Ω , y una de 220 Ω		Todas a ½ Watt.
Pantalla LCD de 16x2 caracteres		Con pines soldados por el usuario.
Tarjeta protoboard pequeña. 400 contactos		

El instrumento armado en la tarjeta protoboard se puede ver en la figura 1. El diagrama esquemático se muestra en la figura 5 del Anexo A2, donde se aprecian claramente cada uno de los componentes listados antes, así como su localización. Para mayor referencia consultar el Tutorial Sobre Arduino (Bañuelos, 2016: p28).

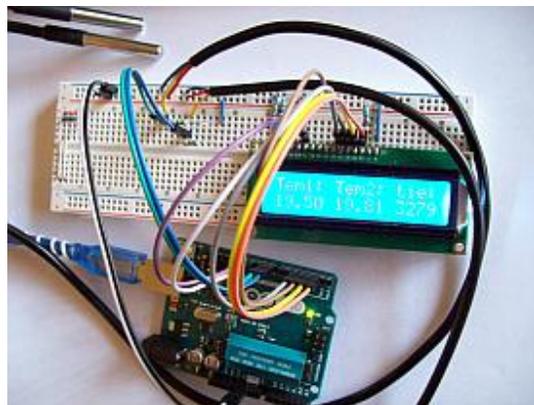


Figura 1. Instrumento armado en la tarjeta protoboard.

4. Equilibrio térmico

En el esquema se aprecia sólo un termómetro, sin embargo, deben incluirse dos termómetros (ver la figura 1). Para tal fin, hay que colocar un segundo termómetro en paralelo (ver figura 6 del Anexo A3), con su terminal DATA conectada al PIN 9.

Una de las fortalezas del sensor DS18B20 radica en que pueden conectarse múltiples de ellos al mismo PIN de datos, sin embargo, aquí no se ha optado por esta modalidad debido a que la complejidad del sketch aumentaría innecesariamente.

Los materiales para implementar el sistema térmico y su frontera aislante son: dos vasos de unicel de 1 L con sus respectivas tapas, un vaso pequeño de plástico o metal con capacidad de 200 ml, cuya pared sea muy delgada. Una porción de 200 ml de agua hirviendo y otra porción de 200 ml de agua a punto de congelación.

Software

- La programación de la tarjeta Arduino se realiza utilizando el Arduino IDE, con el Sketch mostrado en el Anexo A1. Para reducir el número de datos registrados, el programa sólo registra un dato cada vez que detecta una variación de 0.5 °C. En esta práctica se ha empleado el IDE 1.8.2 (o superior) para Windows, pero cada experimentador está en libertad de utilizar la versión de su preferencia; la que puede descargar desde el sitio oficial www.arduino.cc.
- Es requisito indispensable instalar las bibliotecas 1-wire para la comunicación con el sensor y la del manejo del sensor DS18B20 (Bañuelos, 2017: p28).
- Para graficar los datos medidos, debe disponerse de algún tipo de hoja de cálculo como la que viene incluida en todo tipo de paquete de Ofimática comercial o libre.

GUÍA PARA EL PROFESOR: Se sugiere instalar el Sketch mostrado en el Anexo A1, así como las bibliotecas, en cada una de las tarjetas Arduino, antes de que se realice la práctica.

DESARROLLO

Enseguida se enumeran los pasos a seguir.

GUÍA PARA EL PROFESOR: La numeración proporciona una guía a los estudiantes para evitar la pérdida de continuidad, y por otro lado, es una referencia del profesor; por ejemplo, para que pueda preguntar si el alumnado ya terminó algún paso en particular.

1. Montar el sistema experimental mostrado en la figura 2. Se requiere realizar dos perforaciones a las tapas de los vasos grandes de unicel, para poder introducir los sensores de temperatura. El montaje consiste esencialmente de dos porciones de agua, una hirviendo y otra a punto de congelación, que están en contacto térmico. La frontera del sistema se implementa con dos vasos de unicel de 1 litro, apilados, para aumentar su grosor y hacer más eficiente el aislamiento. Dentro de esta frontera se vierte la porción de 200 ml de agua a punto de congelación y dentro de ésta última se sumerge, a la brevedad, el vaso pequeño que contiene los 200 ml de agua hirviendo, para conseguir el contacto térmico.

GUÍA PARA EL PROFESOR: Una alternativa para la frontera es un tortillero de unicel, con una pared diatérmica colocada dentro de él, tal que divida la sección circular interior en dos.

4. Equilibrio térmico

2. Cerrar con ambas tapas la frontera (vasos), inmediatamente después de que el sistema de líquidos está listo. Además, por las tapas pasarán los dos sensores que miden la temperatura en cada uno de



Figura 2. Agua hirviendo y agua a punto de congelación, dentro de la frontera de unicel.



Figura 3. Frontera aislante (con tapas) a través de la cual pasan los dos sensores de temperatura.

los dos líquidos (ver Figura 3). Seguidamente hay que poner en marcha el instrumental Arduino desde el IDE.

3. Monitorear continuamente las dos temperaturas, que se presentan en la pantalla LCD. La medición de estos datos y del tiempo, se realiza de forma autónoma con el instrumento, así como también la presentación de estas mediciones en el Monitor Serie del Arduino IDE.

4. Cuando las temperaturas mostradas en la pantalla LCD son iguales, o aproximadamente iguales, se puede considerar que el sistema llegó al equilibrio.

5. En este momento hay que utilizar el apuntador del ratón para seleccionar los datos que se presentan en el Monitor Serie; teclear Ctrl+Copy para copiarlos y después pegarlos a la hoja de cálculo, donde se procesarán para su análisis.

RESULTADOS

Los datos copiados desde el Monitor Serie del Arduino IDE se grafican en la hoja de cálculo, para observar el comportamiento del sistema. El gráfico resultante, obtenido por el autor, se muestra en la figura 4, en donde se observa cómo desciende gradualmente la temperatura del agua que inicialmente estaba hirviendo y cómo se incrementa la temperatura del agua que inicialmente estaba a punto de congelación. El tiempo total requerido para alcanzar el equilibrio térmico, en este caso, fue de 3 horas y 7 min.

4. Equilibrio térmico

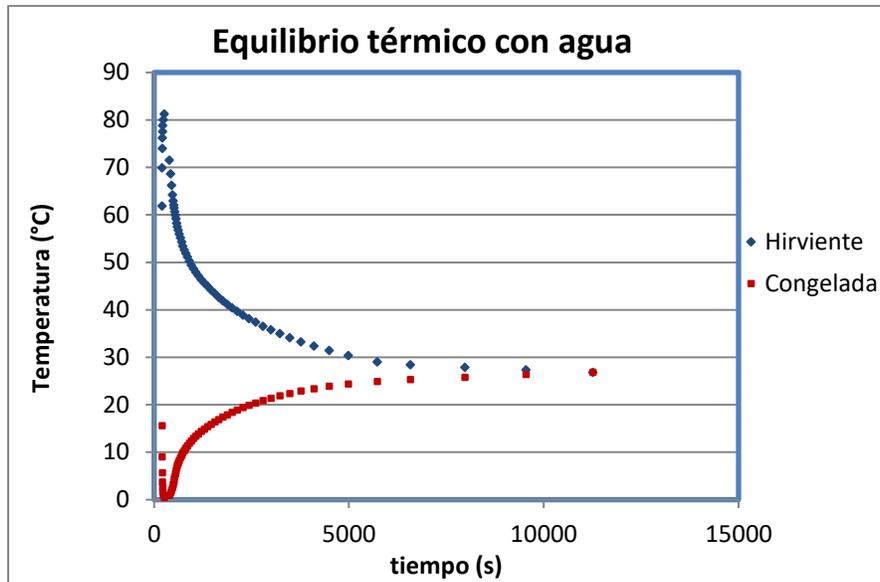


Figura 4. Comportamiento del sistema térmico.

En la parte inicial del proceso se observa como ambos termómetros van alcanzando el equilibrio térmico con el agua en la que se encontraban sumergidos. Finalmente, la temperatura de equilibrio alcanzada es de 26.8 °C.

Esta es sólo una muestra de los posibles resultados a obtener, debido a que el gráfico que obtenga el alumnado puede diferir a causa de diversos factores, como la diferencia de temperaturas entre el agua caliente y la fría a punto de congelación.

DISCUSIÓN Y CONCLUSIONES

Discusión de resultados:

El comportamiento de los resultados obtenidos que se presentan en la figura 4, concuerda con lo reportado en la literatura (Pathare, 2015). Se sugiere que el alumnado mida la *masa* de cada una de las dos porciones de agua, e investigue el valor del *calor específico* del agua, con la intención de utilizarlos en el cálculo del calor $Q_{perdido}$ y del calor Q_{ganado} de las porciones de agua respectivas. Para este cálculo debe utilizarse la ecuación,

$$Q = mc\Delta t \quad (2)$$

en donde Δt se determina auxiliándose del gráfico resultante, figura 4, y de los datos medidos. Los valores numéricos calculados de $Q_{perdido}$ y Q_{ganado} deben compararse para determinar si se cumple la ecuación 1.

Se sugiere que el alumnado responda las siguientes preguntas:

- ¿Se cumple la ecuación 1?
- ¿Cómo es la ley que gobierna el descenso de la temperatura en el agua hirviente?
- ¿Cómo es la ley que gobierna el ascenso de la temperatura en el agua a punto de congelación?
- ¿Cómo será el equilibrio térmico entre dos materiales con capacidades térmicas idénticas?
- ¿Cómo será el equilibrio térmico entre dos materiales con capacidades térmicas diferentes?

4. Equilibrio térmico

CONCLUSIONES

Esta práctica ha permitido observar el comportamiento del intercambio de calor (transferencia de energía) que se presenta entre dos muestras de agua, a temperaturas inicialmente diferentes. El agua hirviente cede calor al agua a punto de congelación, hasta que se equilibra la diferencia térmica. Como trabajo adicional, se propone que el alumnado realice esta práctica con alguna otra combinación de líquidos como aceite, atole, etc., además del agua.

REFERENCIAS

Bañuelos, M. A. (2017). Tutorial sobre Arduino. México: ICAT – UNAM.

García-Colín, L. (2008). De la Máquina de vapor al cero absoluto. México: Editorial FCE.

Pathare, S. et al (2015). Understanding thermal equilibrium through activities. Phys. Educ. vol. 50, no. 2, pp 146-158.

Tippens, P. (2011). Física, Conceptos y aplicaciones. México: Editorial McGraw-Hill Interamericana.

4. Equilibrio térmico

ANEXOS

A1.- Programa (Sketch) que debe grabarse en el Arduino IDE para medir de manera independiente dos temperaturas diferentes.

```
/* Programa que implementa la lectura en dos sensores de temperatura DS18B20 y
 * despliega el resultado en un display LCD y en el Monitor Serie */
```

```
// Se requiere la biblioteca de uso del display
#include <LiquidCrystal.h>
```

```
// Se requiere la biblioteca de comunicación 1-wire, y la del sensor DS18B20
#include <OneWire.h>
#include <DallasTemperature.h>
```

```
// Se selecciona el pin 8 y 9 para lectura de los sensores
#define ONE_WIRE_BUS 8 //Sensor 1
#define TWO_WIRE_BUS 9 //Sensor 2
```

```
// Se activa un Controlador de Protocolo OneWire, para cada sensor.
OneWire oneWire(ONE_WIRE_BUS);
OneWire twoWire(TWO_WIRE_BUS);
```

```
// Se asigna a cada Controlador de Protocolo OneWire cada sensor (DallasTemperature)
// y se le asigna el nombre de: sensors1 y sensors2
DallasTemperature sensors1(&oneWire);
DallasTemperature sensors2(&twoWire);
```

```
// Se definen los pines que se comunican con el display
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
// Se definen características de las variables utilizadas
unsigned long tiempo;
```

```
float iniTemp1;
float finTemp1;
float difeTemp1;
```

```
float iniTemp2;
float finTemp2;
float difeTemp2;
```

```
// Inicialización
void setup() {
```

```
//Inicializa la biblioteca que se denominó como: sensors1 y sensor2.
sensors1.begin();
sensors2.begin();
```

4. Equilibrio térmico

```
// Especifica características del LCD, es de 2 renglones y 16 caracteres
lcd.begin(16, 2);

// Manda un letrero al display LCD
lcd.print("Tem1: Tem2: tie:");

// Velocidad de la comunicación serial
Serial.begin(9600);
}

// Programa principal
void loop() {

// Prepara los sensores para tomar las lecturas
sensors1.requestTemperatures();
sensors2.requestTemperatures();

// Lee la temperatura del sensor 0 y 1
iniTemp1=sensors1.getTempCByIndex(0);
iniTemp2=sensors2.getTempCByIndex(0);

// Bucle en el que se evalúa el cambio de la temperatura medida
do
{
// Espera un segundo
delay(1000);

// Prepara los sensores para tomar las lecturas
sensors1.requestTemperatures();
sensors2.requestTemperatures();

// Lee de nuevo la temperatura del sensor 0 y 1
finTemp1=sensors1.getTempCByIndex(0);
finTemp2=sensors2.getTempCByIndex(0);
// Toma la lectura de tiempo en segundos, en ese instante.
tiempo=millis()*0.001;

// Se posiciona el cursor en el renglón 0, caracter 1; de la pantalla LCD
// Nota: el renglón 1 es el segundo de abajo hacia arriba
lcd.setCursor(0, 1);
// Despliega en LCD la temperatura del sensor 0
lcd.print(finTemp1);

// Imprime un espacio
lcd.print(" ");

// Despliega en LCD la temperatura del sensor 1
lcd.print(finTemp2);
```

4. Equilibrio térmico

```
// Imprime un espacio
lcd.print(" ");

//Despliega en LCD el tiempo (en segundos)
lcd.print(tiempo);

// Calcula la diferencia de temperaturas
difeTemp1=abs(iniTemp1-finTemp1);
difeTemp2=abs(iniTemp2-finTemp2);

} while (difeTemp1 < 0.5 || difeTemp2 < 0.5);

// Cuando se detecta una variación en la temperatura de medio grado, las últimas // lecturas
// de temperatura se mandan al Monitor Serie junto con el tiempo en el que se //registraron.

// Envía un letrero al Monitor serie
Serial.print("Temp1.: ");
// Envía el dato de la temperatura1 del sensor 0, al Monitor.
Serial.print(finTemp1);
// Inserta un espacio después del dato de temperatura1.
Serial.print(" ");
// Escribe el símbolo ""
Serial.print((char)186);
// Inserta la letra C.
Serial.print("C");

// Inserta dos espacios después del dato de temperatura1.
Serial.print(" ");
// Envía un letrero al Monitor serie
Serial.print("Temp2.: ");
// Envía el dato de la temperatura2 del sensor 1, al Monitor.
Serial.print(finTemp2);
// Inserta un espacio después del dato de temperatura2.
Serial.print(" ");
// Escribe el símbolo ""
Serial.print((char)186);
// Inserta la letra C.
Serial.print("C");

// Inserta dos espacios después del dato de temperatura2.
Serial.print(" ");
// Envía un letrero al Monitor serie
Serial.print("tiempo: ");
// Envía el dato del tiempo, al Monitor.
Serial.print (tiempo);
// Inserta un espacio después del tiempo.
```

4. Equilibrio térmico

```
Serial.print(" ");  
// Inserta la unidad s, con salto a línea siguiente  
Serial.println("s");  
}
```

A2.- Diagrama esquemático del circuito electrónico y las conexiones al Arduino

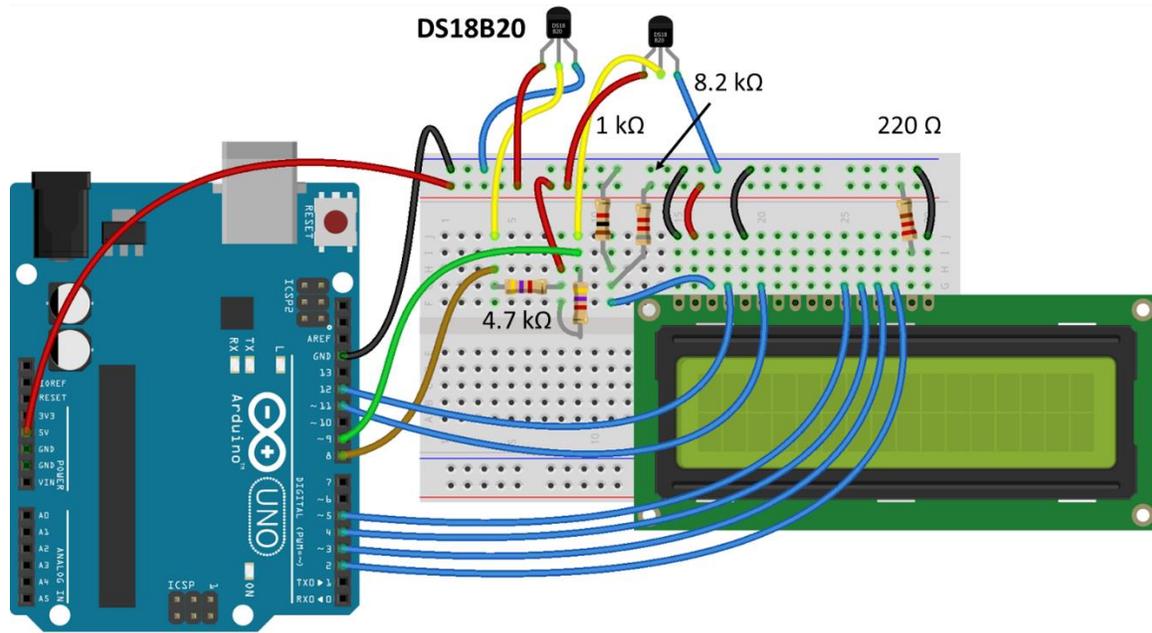


Figura 5. Esquema con las conexiones requeridas en el armado del instrumento.

A3.- Conexión en paralelo de los dos sensores de temperatura.

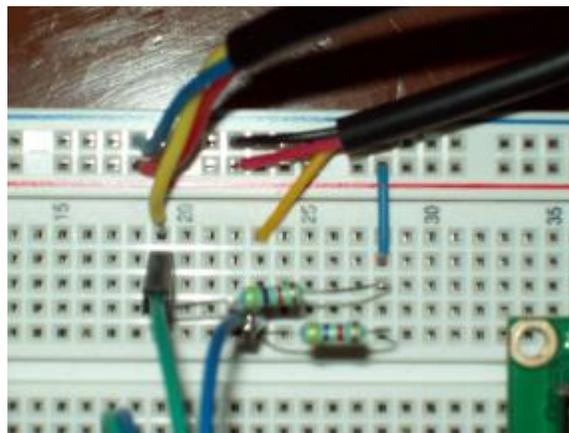


Figura 6. Conexión en paralelo de los dos sensores, una terminal DATA va al pin 8 y la otra al pin 9.

5. TRANSFERENCIA DE CALOR

Responsable: Ismael Rivera Jiménez CCH-Sur	Participantes: Miguel Á. Bañuelos Saucedo ICAT	Revisión: Eduardo José Vega Murguía ICAT
--	--	--

Ubicación curricular

Colegio de Ciencias y Humanidades

Física I

Unidad 3. Fenómenos Termodinámicos

INTRODUCCIÓN

Cuando se le suministra calor a una sustancia u objeto, lo que se espera es un aumento en su temperatura, pero se pueden observar situaciones en las cuales, a pesar de que se le suministre calor a esta sustancia o cuerpo, no se nota un cambio en su temperatura. Cuando se identifican estos procesos lo que ocurre es un **cambio de fase**, esto quiere decir que la sustancia ésta cambiando su forma. Los cambios de fase en una sustancia se pueden presentar en un proceso llamado de **fusión**. En este caso, si la sustancia en su estado inicial es sólida, cuando se le suministra calor, pasa a un estado líquido. También se presenta un cambio de fase en el proceso de **evaporización**, en el cual una sustancia pasa de su estado líquido o un estado gaseoso. Existen otros procesos de cambio de fase, como el de sublimación que consiste en que una sustancia pasa de un estado sólido a un estado gaseoso.

En esta actividad experimental, se pretende identificar los cambios de fase de una sustancia (agua para este caso), el calor necesario para estos cambios de fase y el calor necesario para el cambio de temperatura en la sustancia; así como identificar en una gráfica en qué región o zona se dan estos cambios de fase y de aumento en la temperatura. Para lograr alcanzar los propósitos planteados, es necesario que los alumnos conozcan y apliquen los conceptos de: caloría, equivalente mecánico del calor, calor específico, calor latente y formas de transferencia del calor (conducción, convección y radiación). Por lo que se hace necesaria una investigación de los conceptos y de los modelos matemáticos que los describen. Esta es la herramienta para la interpretación y el análisis de los datos y de la gráfica a obtener.

OBJETIVOS

Objetivos de aprendizaje:

- Obtener la gráfica de temperatura contra tiempo.
- Describir de manera escrita el comportamiento de la gráfica.
- Determinar los puntos de fusión y de ebullición del agua.
- Determinar la energía suministrada en zonas de la gráfica donde se den cambios en la temperatura.
- Calcular la energía suministrada en los intervalos de temperatura dados.

5. Transferencia de calor

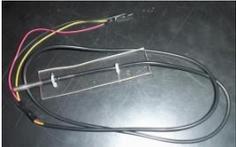
Objetivos de operación

- Utilizar los programas de Arduino, CoolTerm, Excel y Word para la obtención y el manejo de los datos.
- Armar el diagrama eléctrico del apéndice 2.

MATERIAL

Descripción	Imagen	Comentario
Balanza granataria		Para medir la masa del hielo a ocupar.
Mechero de Bunsen		Se utiliza como fuente de calor.
Vaso de precipitado de 500 ml		Contenedor para el hielo.
Soporte universal y accesorios		Para colocar el vaso con el hielo al cual se le suministrara calor con el mechero de bunsen.

5. Transferencia de calor

Descripción	Imagen	Comentario
Accesorios para soporte universal		Disposición del equipo de laboratorio para el proceso experimental.
Sensor de temperatura DS18B20 con base de acrílico (la base es opcional)		Este sensor medirá la temperatura de la sustancia desde su estado sólido hasta el estado líquido en su punto de evaporización.
Arduino UNO		Tarjeta de adquisición de datos a la cual se instala el programa del sensor de temperatura.
Computadora portátil		Con la computadora se instala el programa en la tarjeta Arduino además de suministrar de energía a la tarjeta
Protoboard y una resistencia de 4.7 kΩ @ ½ W		Para realizar las conexiones del sensor de temperatura con la resistencia para su protección.
Cables de conexiones jumpers macho-macho. Cable USB A macho a B macho		Se utiliza un cable USB a y b con el que se transmite y recibe la información a la computadora (datos), además se necesitan cables de conexión para el Arduino, el protoboard y el sensor

5. Transferencia de calor

Software

Para llevar a cabo este proceso experimental se requiere tener instalados en la computadora el siguiente software:

- Programa Arduino IDE 1.8.2 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>.
- Programa CoolTerm 1.4.7 o superior. Disponible en: <http://freeware.the-meiers.org/>.
- Excel y Word, o equivalente.

DESARROLLO

El experimento consiste en calentar una masa de hielo y registrar la variación de la temperatura durante el proceso de fusión y hasta alcanzar el punto de ebullición.

El hielo se colocará en un vaso de precipitado, montado sobre el mechero con ayuda del soporte universal, de acuerdo con la figura 1. Se requiere una pinza de sujeción para detener el sensor de temperatura.



Figura 1. Arreglo experimental con el equipo de laboratorio.

Es necesario conocer la cantidad de hielo, por lo que se deberá colocar una cantidad entre 300 y 400 g de hielo en el vaso de precipitado. Para ello es necesario pesar primero el vaso de precipitado vacío, y luego con el hielo (ver la figura 2). La diferencia de estos dos valores representa la masa de hielo.

5. Transferencia de calor



Figura 2. Medir la masa del hielo.

Después de colocar el vaso de precipitado con hielo sobre el mechero, se deberá introducir en el hielo el sensor de temperatura, cuyas conexiones se muestran en el Anexo 2; de forma tal, que el montaje experimental quede como se muestra en la figura 3. La tarjeta Arduino deberá conectarse a la computadora personal. Para la adquisición de los datos se utilizará el programa CoolTerm. Para conocer los detalles del proceso de registro de datos utilizando este programa se sugiere consultar el procedimiento descrito en la práctica 2 Movimiento rectilíneo de este mismo manual. Para utilizar el sensor de temperatura DS18B20 se requiere la instalación de dos bibliotecas, cuyos detalles de su proceso de instalación se pueden consultar en el Tutorial sobre Arduino (Bañuelos, 2017).

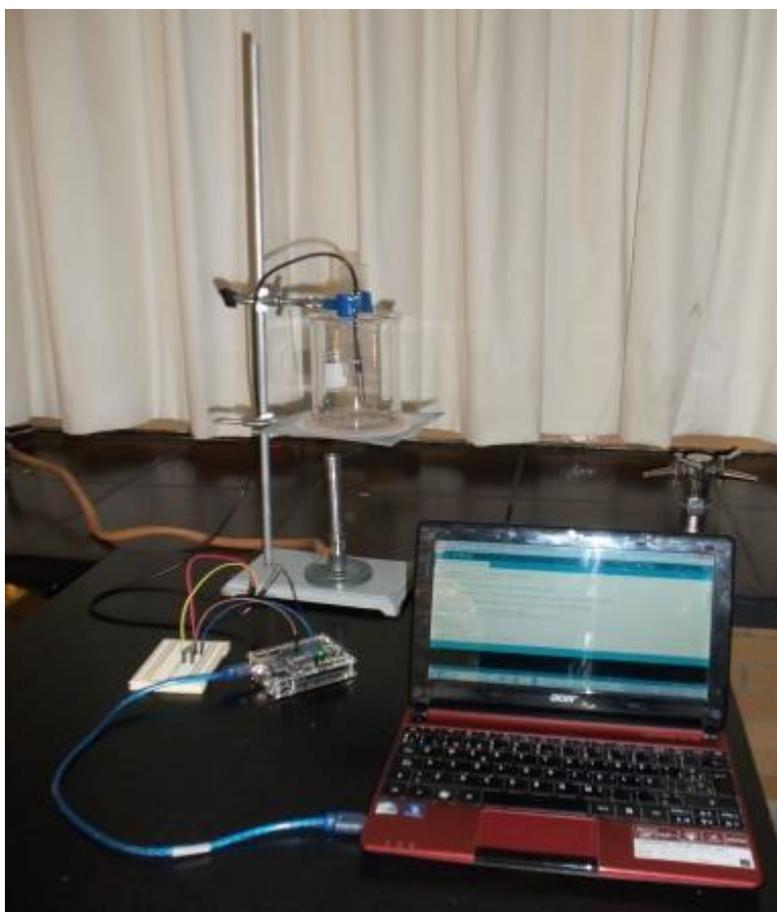


Figura 3. Montaje del dispositivo experimental y conexiones electrónicas.

5. Transferencia de calor

Finalmente, antes de encender el mechero, deberá iniciarse el registro de datos del Arduino con el programa CoolTerm. En la figura 4, se muestra parte del proceso experimental, en ella se puede apreciar el sensor de temperatura dentro de la sustancia (agua), y las conexiones electrónicas de éste y la tarjeta Arduino. Los datos son enviados a la computadora a través del puerto serial de ésta, a razón de un dato cada tres segundos. El experimento se puede dar por finalizado a los pocos minutos de iniciado el proceso de ebullición.

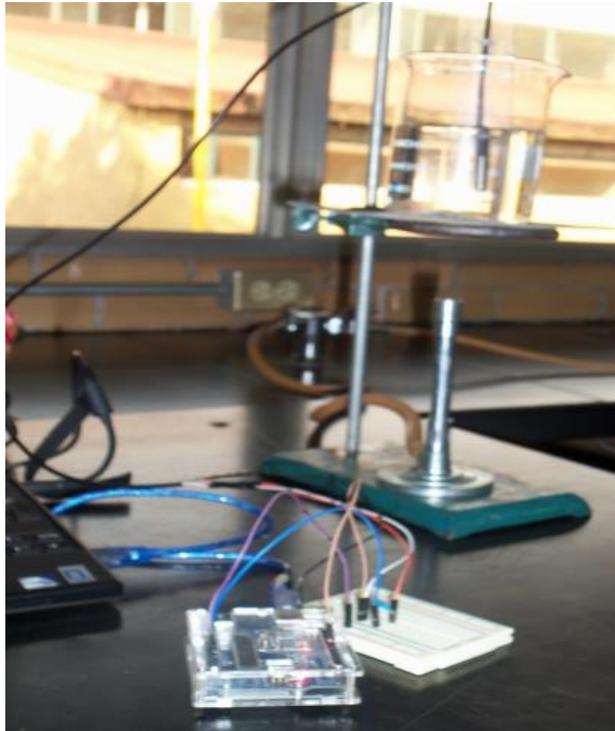


Figura 4. Dispositivo experimental en el proceso de toma de datos.

RESULTADOS

El archivo de datos obtenido se puede abrir en un programa de hoja de cálculo para generar una gráfica semejante a la de la figura 5. Con esta gráfica, y utilizando las ecuaciones:

$$Q = c_e m \Delta T \quad ; \quad Q = mL \quad \text{y} \quad 1 \text{ cal} = 4.184 \text{ J}$$

donde:

- Q es el calor
- c_e es el calor específico de la sustancia
- m es la masa de la sustancia
- ΔT es el cambio en la temperatura
- L es el calor latente ya sea de fusión o de vaporización,

se pueden hacer los cálculos necesarios para determinar cuánto calor se necesita para el cambio de fases. También es posible calcular el calor necesario, que se ha suministrado al sistema, para elevar

5. Transferencia de calor

la temperatura, por ejemplo, de 45 °C a 50 °C, o en cualquier otro intervalo donde la temperatura esté cambiando.

Es posible hacer cambios de unidades del calor utilizando el equivalente mecánico del calor.

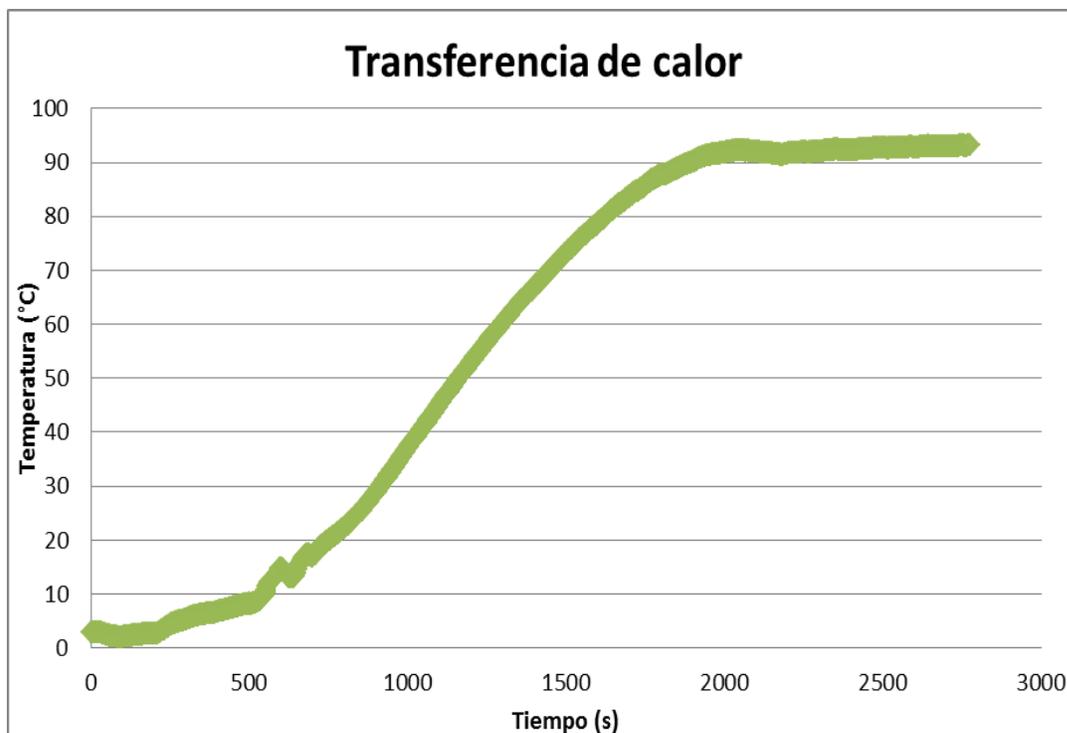


Figura 5. Gráfica de temperatura contra tiempo.

DISCUSIÓN Y CONCLUSIONES

En la gráfica que se obtiene, los alumnos deberán identificar que el agua hierve aproximadamente a 94 °C y no a 100 °C, como está escrito en los libros. Su sugiere abrir la discusión sobre las posibles causas de este resultado.

Usando las ecuaciones de calor latente y calor específico, ¿cuánto calor se ha suministrado a la sustancia? Por ejemplo, si tenemos 400 g de agua y el calor específico del agua es de 4.184 J/g °C, para un cambio en la temperatura de 45 °C a 50 °C se tiene:

$$Q = 4.184 \frac{J}{g^{\circ}C} 400g(50^{\circ}C - 45^{\circ}C) = 8368 J$$

Algunos otros cálculos y comentarios que se podrían hacer son:

- Calcular el calor latente de fusión y de evaporización.
- La cantidad de energía suministrada por unidad de tiempo cuando cambia la temperatura.
- Dar los resultados en Joules y en calorías.

5. Transferencia de calor

- Ver si se le puede asociar una ecuación de tendencia a la zona donde se observa un incremento en la temperatura en la gráfica.
- Comentar en qué parte del proceso experimental se observan los procesos de transferencia de calor.

Conclusiones

Algunas de las conclusiones, a las que pueden llegar los alumnos son: que a las sustancias cuando se les suministra calor, se pueden identificar las fases por las que pasan éstas; que es posible calcular la energía necesaria para que ocurran estos cambios de fase; y finalmente, que no en todo el proceso de calentamiento, necesariamente la temperatura está cambiando.

REFERENCIAS

Bañuelos, M. A. (2017). Tutorial sobre Arduino. México: ICAT-UNAM.

Bueche, F. (1996). *Fundamentos de Física: Tomo I*. Sexta edición. México: Mc Graw Hill.

Giancoli, D. C. (2006). *Física principios con aplicaciones*. Sexta edición. México: Pearson Educación.

Romanelli, L., Fendrik A. (2001). Física. México: Pearson Educación.

Tippens, P. (2001). *Física: Conceptos y aplicaciones*. Séptima edición. México: Mc Graw Hill.

Wilson, J., Buffa, A., Lou, B. (2007). *Física*. Sexta edición. México: Pearson Educación.

Zitzewitz, P. (2004). *Física: Principios y problemas*. México: Mc Graw Hill.

5. Transferencia de calor

ANEXOS

1.- Programa del sensor de temperatura que se graba en el Arduino para hacer el experimento.

```
// Programa sensor de temperatura

#include <OneWire.h> //Se importan las bibliotecas
#include <DallasTemperature.h>

#define Pin 8 //Se declara el pin donde se conectará y recolectarán los datos
OneWire ourWire(Pin); //Se establece el pin declarado como bus para la comunicación OneWire

DallasTemperature sensors(&ourWire); //Se instancia la biblioteca DallasTemperature
unsigned long time = 0; //declaración de tiempo
unsigned long tim = 0; //declaración de tiempo dividido entre mil

void setup() {
  //delay(1000);
  Serial.begin(9600);
  sensors.begin(); //Se inician los sensores
  Serial.println(" t Tem"); //se da espacio e imprime las palabras tiempo "t" y temperatura "Tem"
  Serial.println(" s C"); //imprime las unidades de tiempo y temperatura en el monitor serie
}

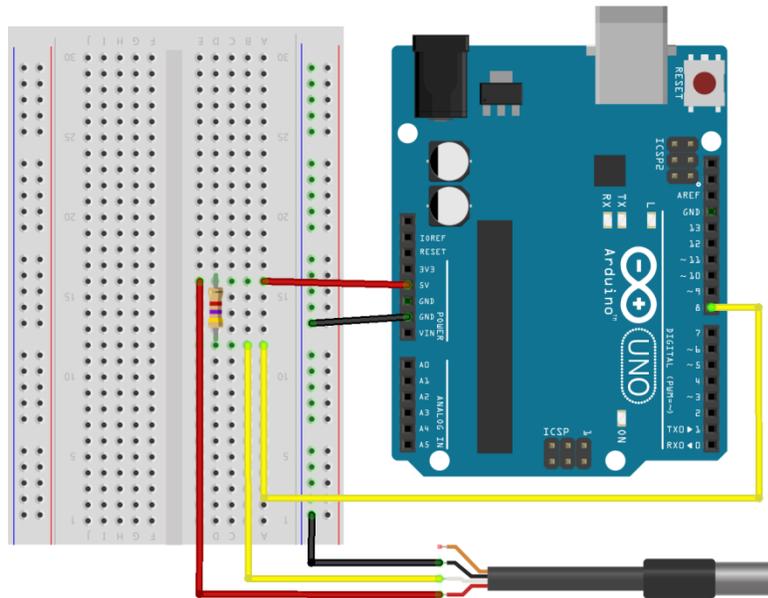
void loop() {
  sensors.requestTemperatures(); //Prepara el sensor para la lectura
  Serial.print(" "); //espacio para los valores de tiempo
  time=millis(); //Obtiene el tiempo en milisegundos que lleva encendido el Arduino
  tim=time/1000; // Calcula los segundos que lleva encendido el Arduino
  Serial.print (tim); //imprime tiempo en monitor serie

  Serial.print (" "); //se da espacio entre "s" y el valor de la temperatura
  Serial.println(sensors.getTempCByIndex(0)); //lee e imprime la temperatura en grados Celsius

  delay(3000); //Se produce un lapso de 3 segundos antes de la próxima lectura
}
```

5. Transferencia de calor

2.- Diagrama esquemático del circuito electrónico que se conecta al Arduino



6. PRESIÓN ATMOSFÉRICA Y ALTURA

Responsable: Sergio Alejandro Carrillo Araujo CCH Sur	Participantes:	Revisión: Eduardo José Vega Murguía ICAT
---	----------------	--

Ubicación Curricular

Colegio de Ciencias y Humanidades

Física I, Área de Ciencias Experimentales

Unidad 1. Introducción a la Física

Unidad 3. Energía: fenómenos térmicos, tecnología y sociedad

Resumen: Se presenta una estrategia para medir presión aplicando la tarjeta Arduino Nano, la tarjeta Arduino UNO, y el sensor de presión BMP180. Se emplean algunos sensores para llevar a cabo una serie de mediciones y abordar los conceptos de presión atmosférica, temperatura y altura (altitud), con estudiantes de tercer semestre del Colegio de Ciencias y Humanidades. El uso de microcontroladores y sensores en el bachillerato resulta motivante para los alumnos, ya que a la vez que amplía su horizonte cultural fomenta su curiosidad para entender su funcionamiento.

Palabras Clave: Presión, Atmosfera, Sensores, Arduino

INTRODUCCIÓN

El concepto de presión se contempla en diversos aprendizajes del programa de Física del Colegio (CCH UNAM, 2003), y es un concepto aparentemente fácil de comprender dada la sencillez de su definición (la fuerza ejercida por un fluido por unidad de área). Sin embargo, cuesta un poco de trabajo hacer comprender a los alumnos que vivimos inmersos en una capa de aire y que esta se encuentra ejerciendo fuerza sobre cada punto de nuestro cuerpo, sobre cada superficie, la llamada presión atmosférica. El valor promedio de la presión atmosférica a nivel del mar se considera de $1 \text{ atm} = 101\,300 \text{ Pa}$. Una de las unidades frecuentes para medir la presión también es el mmHg, siendo la equivalencia $760 \text{ mmHg} = 1 \text{ atm}$.

El físico italiano Evangelista Torricelli (1608-1647) diseñó un dispositivo llamado barómetro (Bueche, 1991) que consiste en un tubo de vidrio de un metro de longitud cerrado por un extremo al cual llenó con mercurio y depositó de forma invertida sobre una cubeta llena del mismo material. Al ver hasta dónde descendía la columna de mercurio en el tubo, cuando se alcanzaba el equilibrio con la presión atmosférica, notó que se establecía en 760 mm, con lo que quedó demostrada la existencia de la presión atmosférica y se determinó su valor por primera vez (ver figura 1).

6. Presión atmosférica y altura

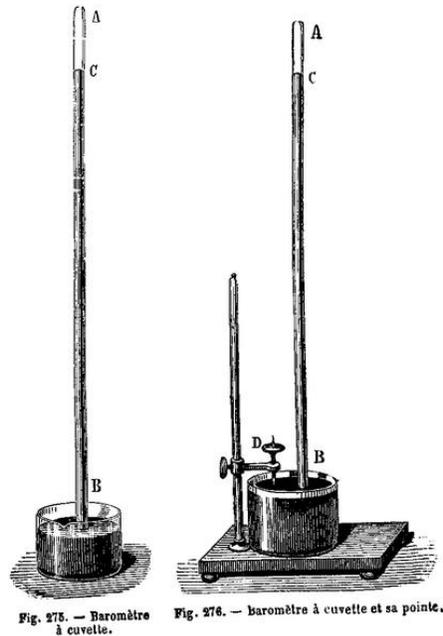


Figura 1. Barómetro de mercurio. Experimento ideado por Torricelli. Fuente: Wikipedia.

La presión tiene la siguiente relación con la temperatura y la altura (Resnick, Halliday y Krane, 2001)

$$P_h = P_0 e^{-\frac{mgh}{kT}}$$

En esta práctica se aborda el concepto de presión atmosférica con un sensor de presión atmosférica BMP180 montado en una placa de Arduino Nano. Se comparan los resultados obtenidos con los sensores y se determinó la diferencia de altura desde el estacionamiento de profesores hasta el edificio que se encuentra a mayor altura del plantel.

OBJETIVOS

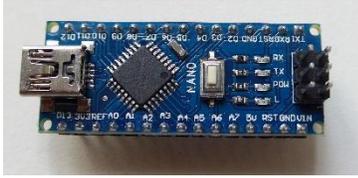
El objetivo general de esta práctica es:

Que el alumno comprenda el concepto de presión atmosférica y determine de forma indirecta la altura de algún lugar en el Colegio o en casa.

Un objetivo particular es la aplicación de sensores con Arduino para medir presión atmosférica y determinar altura de distintos lugares.

6. Presión atmosférica y altura

MATERIAL

Descripción	Imagen	Comentarios
Microcontrolador Arduino Nano		<p>Arduino Nano es de los más sencillos, pequeños y baratos de los Arduinos que se encuentran en el mercado.</p>
Cable de alimentación		<p>Regularmente mini USB a USB.</p>
Pantalla de Cristal Líquido LCD 16x32 (1602A)		<p>Con LED de fondo azul o verde. Hay pantallas sin LED de fondo son más baratas y funcionan casi del mismo modo. Se requiere que el usuario suelde los pines de conexión.</p>
Módulo controlador I2C para LCD		<p>Con este controlador se puede ahorrar pines de conexión en el Arduino ya que solo necesita cuatro, uno de alimentación 5V, tierra (GND), SDA y SCL. Se recomienda su uso.</p>
Mini Protoboard		<p>Solo se necesitarán pocos pines de conexión. Para proyectos cortos es recomendable pues es económica.</p>
Jumpers macho-macho		<p>Se necesitan varios, se recomienda adquirir por lote de 40 cables.</p>

6. Presión atmosférica y altura

Descripción	Imagen	Comentarios
Jumpers macho-hembra		Se necesitan varios, se recomienda adquirir por lote de 40 cables.
Batería de respaldo de 2000 mAh o similar capacidad		Existen diversas marcas. Revisar que estén cargadas.
Sensor de presión BMP180		

Software

- Programa Arduino IDE 1.8.2 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>.
- Sketch: Programa para operar el sensor BMP180 en Arduino Nano y UNO
- Programa Fritzing 0.9.3b o superior. Disponible en <http://fritzing.org/download/>.

Bibliotecas:

- SFE_BMP180.h Disponible en: <https://github.com/adafruit/Adafruit-BMP085-Library>.
- wire.h Para manejo del display LCD mediante comunicación con el módulo I2C (ya instalada con el programa Arduino IDE).

GUÍA PARA EL PROFESOR:

Para el desarrollo de esta práctica se necesita el software Arduino IDE para la programación de instrucciones para el microcontrolador desde una computadora personal.

Se pueden hacer esquemas de ensamblaje con el programa Fritzing que es software abierto. Es recomendable para que los alumnos manejen las conexiones de los sensores primero de forma virtual.

Existen algunas bibliotecas especiales para algunos sensores; por ejemplo, para el BMP180 se requiere SFE_BMP180.h mientras que para el I2C del LCD se requiere la biblioteca wire.h, además se necesita el material indicado más abajo:

Las bibliotecas especiales para el controlador I2C del LCD pueden requerir borrar algunas bibliotecas LCD anteriores, se recomienda tener un respaldo.

6. Presión atmosférica y altura

DESARROLLO

Para llevar a cabo esta actividad se deberán seguir los siguientes pasos:

1. Realizar una breve recopilación de datos sobre el concepto de presión, los tipos de presión y sus aplicaciones.
2. Responder ¿cómo se puede medir la presión atmosférica?
3. Revisar el cuadro de unidades para determinar cómo se puede medir la presión.

	Pascal	Bar	Atmósfera	Milímetros de mercurio	Libras por pulgada ²
	(Pa)	(bar)	(atm)	(mmHg)	(psi)
1 Pa		10^{-5}	9.8692×10^{-6}	7.5006×10^{-3}	1.450377×10^{-4}
1 bar	10^5		0.98692	750.06	14.50377
1 atm	1.01325×10^5	1.01325		760	14.69595
1 mmHg	133.3224	1.333224×10^{-3}	1.315789×10^{-3}		1.933678×10^{-2}
1 psi	6.8948×10^3	6.8948×10^{-2}	6.8046×10^{-2}	51.71493	

4. Instalar el sensor BMP180 y el controlador I2C como se muestra en la figura 2.

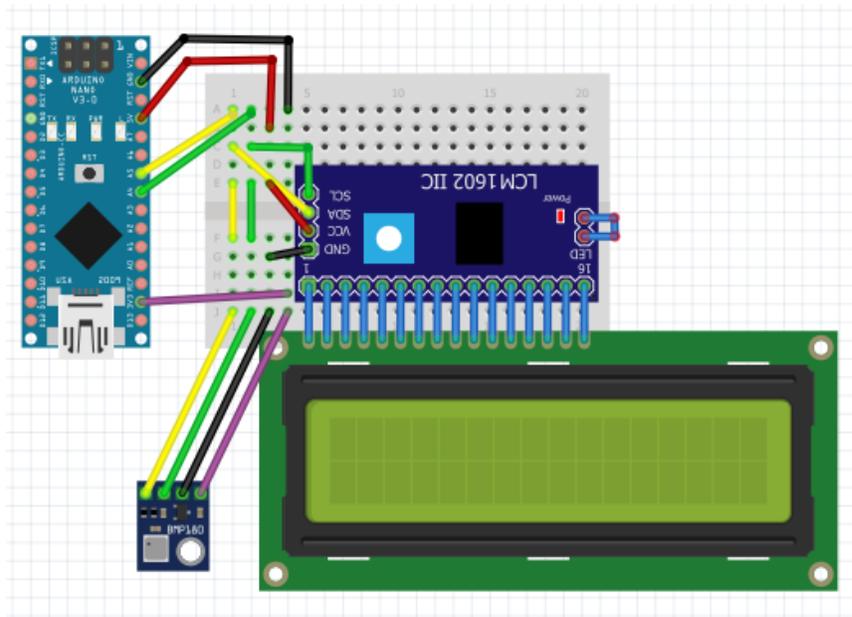


Figura 2. Esquema de conexión del controlador I2C para display LCD y sensor de presión BMP180 en conexión con el microcontrolador Arduino nano.

5. Cargar el programa que se encuentra en el Anexo, cuidando que las bibliotecas sean previamente instaladas para el sensor BMP180 y para el controlador I2C del LCD.
6. Asegurarse de que el sensor lee la presión atmosférica, temperatura y calcula una estimación de la altura.

6. Presión atmosférica y altura

7. Una vez cargado el programa en el microcontrolador Arduino, se puede alimentar con una batería recargable para que sea portable y se pueda trasladar.
8. Comparar la presión atmosférica medida con el sensor, revisar la dependencia de las lecturas con la temperatura y estimar la diferencia de altura con la diferencia de presión marcada por el termómetro.
9. Discutir las diferencias de los resultados con los compañeros.
10. ¿Cuál es la precisión de las mediciones? ¿Cuáles son las características del instrumento?

Resultados y Análisis

GUÍA PARA EL PROFESOR:

El sensor de presión BMP180 además puede medir la temperatura. En este ejemplo se le requirió recuperar una función programada que determina la altura a la que nos encontramos. En esta actividad los alumnos pueden determinar la diferencia de altura entre dos lugares de la escuela como pueden ser el estacionamiento de profesores del Colegio y la Academia de Ciencias Experimentales. Primero se les puede pedir una estimación al tanteo, por ejemplo, en los puntos indicados ellos supusieron una altura entre los 6 y los 10 metros en diferencia.

Luego de armar los sistemas y montar los programas en las placas de Arduino se les puede pedir un comparativo de lecturas entre dos sensores en el laboratorio. En la siguiente tabla aparece un ejemplo de ellas.

Arduino UNO	Arduino Nano
Temp= 25.21 °C Presión= 776.11 mB	Temp= 25.49 °C Presión= 774.71 mB
Altura [m]	Altura [m]
-0.40	-0.60
0.20	0.18
0.49	0.03
0.06	-0.20
0.09	-0.15
0.63	-0.08
0.36	-0.93
0.57	-0.13
0.92	-0.50
0.30	-0.24
0.29	0.40
0.42	-0.50
0.41	0.27
0.49	-0.54
0.48	0.56
0.35	0.69
0.05	0.17
0.73	0.10
-0.03	0.24
0.66	-0.07
0.10	0.26

6. Presión atmosférica y altura

Arduino UNO	Arduino Nano
0.42	0.22
0.58	0.07
0.49	0.08
0.74	0.84
0.66	0.43
0.20	0.24
0.24	0.81
0.28	0.77
0.86	-0.21
1.12	-0.18
0.29	-0.03
0.22	0.99
Promedio = 0.4021 Desv= 0.3033	Promedio=0.0906 Desv= 0.4481

La comparación de los promedios nos sugiere un promedio menor para el sensor BMP180 conectado al Arduino Nano, sin embargo, nos sugiere una desviación estándar mayor que en el sensor conectado al Arduino UNO. Esta comparación es poco concluyente ya que necesitamos repetir más eventos para señalar si simplemente fue una fluctuación o algún sistema resulta más sensible que otro. Lo que es cierto de los datos es que el más inestable en las mediciones fue el sensor conectado al Arduino Nano. Los siguientes resultados obtenidos por los estudiantes sugieren una diferencia pequeña en las presiones, no tanto en la temperatura, y sin embargo las lecturas en la altura contrastaron pues a un equipo les resultaron negativas y a otro positivas.



Figura 3. Lectura de la medición de presión en el estacionamiento de profesores. A la izquierda: medición con Arduino UNO. A la derecha: medición con sensor BMP180 y Arduino Nano. Se observa el contraste de las mediciones en la altura principalmente, luego la temperatura y la presión atmosférica.

Los resultados arrojados en esta medición sugieren una diferencia de alturas de entre 14.53 m y 16.88 m para la subida desde la entrada del estacionamiento de profesores hasta el edificio I, donde se

6. Presión atmosférica y altura

encuentra la academia de Ciencias Experimentales. Se puede considerar una desviación máxima de 0.45 m.

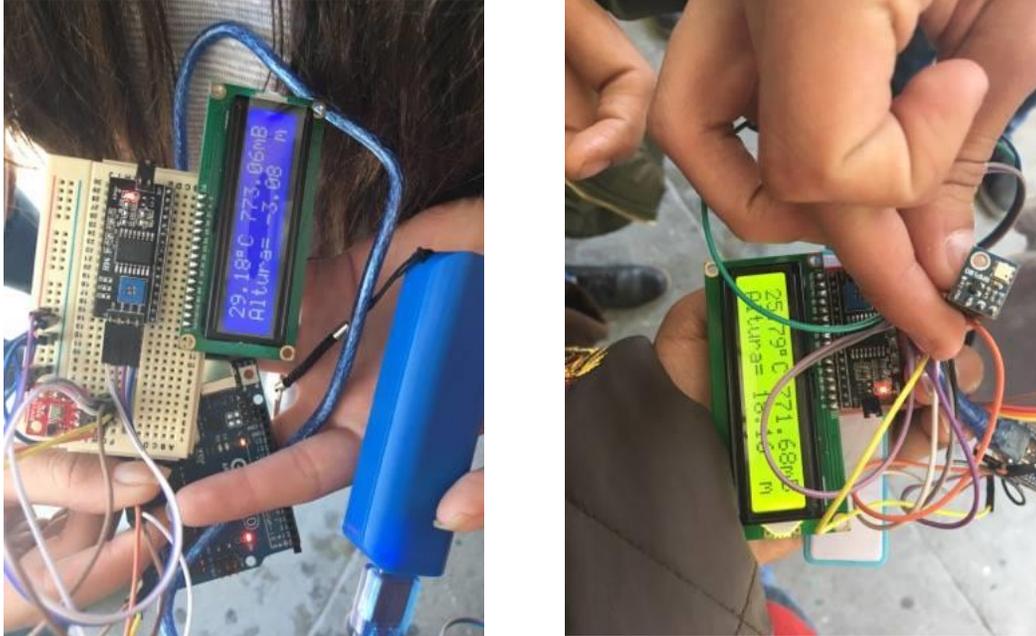


Figura 4. Lectura de datos en la Academia de Ciencias Experimentales Edificio I del Colegio. A la izquierda: datos con Arduino UNO. A la derecha: datos con Arduino Nano. La altura encontrada en UNO es 14.53 m y en Nano es de 16.88 m.



Figura 5. Estudiantes tomando lectura de datos de la presión atmosférica con los Arduinos Nano y UNO en un diseño portátil.

DISCUSIÓN Y CONCLUSIONES

GUÍA PARA EL PROFESOR: Los estudiantes revisaron los conceptos de presión y temperatura motivados en el uso del sensor de presión BMP180 y el microcontrolador Arduino Nano y UNO. Compararon sus resultados entre equipos y determinaron la diferencia de altura con la diferencia de presión registrada por el sensor. Sus resultados aún no son concluyentes pues se les requirió encontrar otra forma de confirmar sus resultados y explicar las diferencias de estos.

La aplicación de sensores eléctricos fomentó en los alumnos el interés en su aplicación y en entender su funcionamiento. Considero que en las prácticas experimentales de física sería ideal implementar el uso de estos equipos de bajo costo para desarrollar estrategias didácticas que proporcionen una formación integral para las nuevas generaciones de alumnos.

Es deseable que el alumno alcance la habilidad de montar una placa Arduino con un sensor de presión con pantalla LCD dentro una sesión de laboratorio. Se debe sugerir una propuesta de aplicación del sensor de presión para responder con evidencias (datos) una pregunta planteada por los alumnos, esto promueve la actitud científica y la innovación en la aplicación de tecnología y conocimiento.

Referencias

Bueche, F. J. (1991). Fundamentos de física. Tomo I. México: McGraw Hill.

CCH UNAM (2003). Plan de estudios física I a IV. Recuperado el 5 de junio de 2019 de http://www.cch.unam.mx/sites/default/files/plan_estudio/mapa_fisica.pdf

Olmo, M. y Nave R. (). La Fórmula Barométrica. Recuperado el 5 de junio de 2019 de: <http://hyperphysics.phy-astr.gsu.edu/hbasees/Kinetic/barfor.html>

Resnick, R., Halliday, D. y Krane, K. (2001). Física Vol. 1. México: CECSA.

6. Presión atmosférica y altura

ANEXO

Programa para operar el sensor BMP180 en Arduino Nano y UNO

```
/******  
***  
* Autor: Sergio Alejandro Carrillo Araujo *  
* CCH Sur UNAM *  
* CC:3.0 *  
* 14/02/2017 Última revisión: 14/02/2017 *  
* *  
* Proyecto: Sensor de presión BMP180 *  
* Estado: Funcional *  
* Material: Arduino Leonardo, UNO, Nano, Display 1602A, *  
* Sensor BMP180, Modulo I2C para LCD *  
* Proposito: Medir presión atmosférica, determinar Altura *  
* *  
* Detalles: Hay que calibrar el sensor, revisar para i2c *  
* https://arduino-info.wikispaces.com/LCD-Blue-I2C#v3 *  
*****  
*/  
#include <Wire.h>  
#include <LiquidCrystal_I2C.h> // Librería especial para LCD disminuye conexiones  
  
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  
  
#include <SFE_BMP180.h>  
//Se declara una instancia de la librería  
SFE_BMP180 pressure;  
  
//Se declaran las variables. Es necesario tomar en cuenta una presión inicial  
//esta será la presión que se tome en cuenta en el cálculo de la diferencia de altura  
double PresionBase;  
  
//Leeremos presión y temperatura. Calcularemos la diferencia de altura  
double Presion = 0;  
double Altura = 0; // Altura en CDMX 2250 m  
double Temperatura = 0;  
  
char status;  
  
void setup()  
{  
  Serial.begin(9600); // Inicia serial  
  
  //Se inicia el sensor y se hace una lectura inicial
```

6. Presión atmosférica y altura

```
SensorStart();

lcd.begin(16,2); // inicia lcd
// -----
for(int i = 0; i < 3; i++)
{
  lcd.backlight();
  delay(250);
  lcd.noBacklight();
  delay(250);
}
lcd.backlight(); // termina backlight

lcd.setCursor(0,0); //Comienza lcd entrada 0,0
lcd.print("Hola Alex!");
delay(1000);
lcd.setCursor(0,1);
lcd.print("CCH SUR UNAM");
delay(8000);

Serial.println("Temperatura[C],Presion[mBar],Altura[m]");
}

void loop()
{
  //Se hace lectura del sensor
  ReadSensor();

  //Se imprimen las variables

  Serial.print(Temperatura);
  Serial.print(", ");

  Serial.print(Presion);
  Serial.print(", ");
  Serial.println(Altura);

  delay(2000);

  lcd.setCursor(0,0);
  // lcd.print("T=");
  lcd.print(Temperatura);
  lcd.print((char)B11011111);
  lcd.print("C ");
  // lcd.print("P=");
  lcd.print(Presion);
  lcd.print("mB");
```

6. Presión atmosférica y altura

```
// lcd.autoscroll();
lcd.setCursor(0,1);
lcd.print("Altura= ");
lcd.print(Altura);
lcd.print(" m");

}

void SensorStart() {

  //Secuencia de inicio del sensor

  if (pressure.begin())
    Serial.println("BMP180 init success");
  else
  {

    Serial.println("BMP180 init fail (disconnected?)\n\n");
    while (1);
  }

  //Se inicia la lectura de temperatura
  status = pressure.startTemperature();
  if (status != 0) {
    delay(status);
    //Se lee una temperatura inicial
    status = pressure.getTemperature(Temperatura);
    if (status != 0) {
      //Se inicia la lectura de presiones
      status = pressure.startPressure(3);
      if (status != 0) {
        delay(status);
        //Se lee la presión inicial incidente sobre el sensor en la primera ejecución
        status = pressure.getPressure(PresionBase, Temperatura);
      }
    }
  }
}

void ReadSensor() {
  //En este método se hacen las lecturas de presión y temperatura y se calcula la altura

  //Se inicia la lectura de temperatura
  status = pressure.startTemperature();
  if (status != 0)
  {
    delay(status);
    //Se realiza la lectura de temperatura
```

6. Presión atmosférica y altura

```
status = pressure.getTemperature(Temperatura);
if (status != 0)
{
  //Se inicia la lectura de presión
  status = pressure.startPressure(3);
  if (status != 0)
  {
    delay(status);
    //Se lleva a cabo la lectura de presión,
    //considerando la temperatura que afecta el desempeño del sensor

    status = pressure.getPressure(Presion, Temperatura);
    if (status != 0)
    {
      //Se hace el cálculo de la altura en base a la presión leída en el Setup
      Altura = pressure.altitude(Presion, PresionBase);
    }
    else Serial.println("error en la lectura de presion\n");
  }
  else Serial.println("error iniciando la lectura de presion\n");
}
else Serial.println("error en la lectura de temperatura\n");
}
else Serial.println("error iniciando la lectura de temperatura\n");
}
//-----FIN-----
```

7. CURVAS DE CALENTAMIENTO Y ENFRIAMIENTO DE LÍQUIDOS CON SENSORES EN ARDUINO

Responsable: Sergio Alejandro Carrillo Araujo CCH-Sur	Participantes:	Revisión: Eduardo José Vega Murguía ICAT
---	----------------	--

Ubicación curricular

Colegio de Ciencias y Humanidades

Física I, Área de Ciencias Experimentales

Unidad 3. Energía: fenómenos térmicos, tecnología y sociedad

Resumen: Se presenta una estrategia para medir temperatura aplicando el microcontrolador Arduino UNO y sensores de temperatura. Se emplearon sensores analógicos y digitales (LM35 y DS18B20) para llevar a cabo una serie de mediciones y abordar los conceptos de temperatura y equilibrio térmico con estudiantes de tercer semestre del Colegio de Ciencias y Humanidades. El uso de microcontroladores y sensores en el bachillerato amplía el horizonte cultural de los estudiantes a la vez que despierta su interés para comprender su funcionamiento y a la física básica.

Palabras clave: Temperatura, Termómetro, Sensores.

INTRODUCCIÓN

En el Colegio de Ciencias y Humanidades, el concepto de temperatura se aborda en la tercera unidad del programa de estudios de física I (CCH UNAM, 2003). Se encuentran incorporados los temas de equilibrio térmico, temperatura e intercambio de energía interna. Los aprendizajes estipulados en el programa indican que el alumno debe:

- Comprender los conceptos de equilibrio térmico, temperatura y calor.
- Describir los cambios de temperatura producidos por intercambio de energía, su relación con la energía interna y emplear el modelo de partículas para explicarlos.

El concepto de temperatura se puede tratar con los alumnos desde varios posibles enfoques, uno que se explica como consecuencia de interacciones microscópicas (teoría cinética de los gases) y que se relaciona con la energía cinética promedio de las partículas que componen a un objeto. La expresión de la energía interna sintetiza la relación con la temperatura para el caso de un gas ideal monoatómico:

$$\langle E \rangle = \frac{3}{2}NkT$$

Donde E es la energía cinética, N el número de partículas, k la constante de Boltzmann y T la temperatura. Esta relación se puede conectar de forma simple con la expresión de los gases ideales

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

que suele manejarse en los diferentes libros del nivel de bachillerato (Bueche, 1991), (Resnick, Halliday y Krane, 2001).

Otro enfoque parte de las manifestaciones macroscópicas observadas a través del cambio producido en las propiedades de los cuerpos al estar en interacción. Regularmente se entiende a la temperatura como la propiedad que comparten los objetos al alcanzar el equilibrio térmico¹, de este principio se deduce su existencia como una propiedad de los objetos compuestos por partículas. Es interesante el enfoque axiomático de la termodinámica que se discute en algunos textos (Martínez, 2012).

Una definición más formal, pero nada intuitiva para los estudiantes de bachillerato, es la que relaciona la rapidez de cambio parcial de la entropía (S) de un sistema con respecto a su energía interna (U), considerando el volumen (V) y el número de partículas (N) constante.

$$T = \frac{1}{\left(\frac{\partial S}{\partial U}\right)_{N,V}}$$

Esta expresión es una mejor aproximación al concepto de temperatura pues toma en cuenta la variedad de formas de energía y los modos colectivos de sistemas; sin embargo, los conceptos físicos y las matemáticas empleadas aquí resultan difíciles de aplicar en nivel de bachillerato.

En esta estrategia se proponen algunas prácticas para entender el concepto de equilibrio térmico y temperatura. Nos apoyamos en el uso de equipo simple de laboratorio e incorporamos sensores con el microcontrolador Arduino.

La temperatura ha sido un concepto físico que ha tardado mucho tiempo en refinarse y comprenderse. El uso de microcontroladores y sensores para medir la temperatura puede contribuir a un estudio con toma y manejo de datos en tiempo real. Esta forma de tratar el concepto aumenta el interés de los alumnos y los conduce al desarrollo de habilidades y al entendimiento de este tema desde una perspectiva diferente, un tanto moderna.

OBJETIVOS

Realizar una estrategia para estudiar los conceptos de temperatura, calentamiento y enfriamiento de sustancias, empleando microcontroladores Arduino y sensores de temperatura analógicos y digitales. Como objetivos particulares está la revisión de los conceptos de temperatura, equilibrio térmico y la aplicación de sensores.

¹ Se dice que dos objetos están en equilibrio térmico entre sí cuando el valor de su temperatura es el mismo o cuando el intercambio neto de energía entre ellos es cero.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

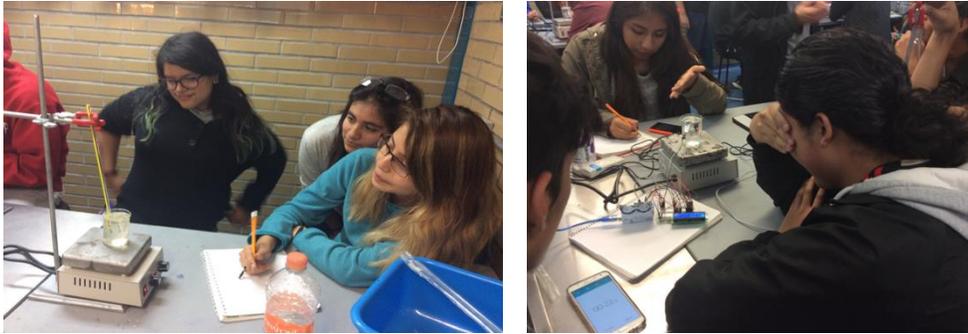
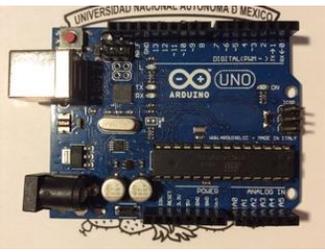
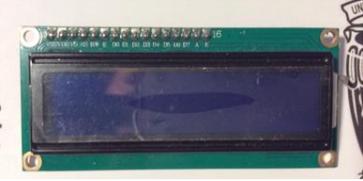
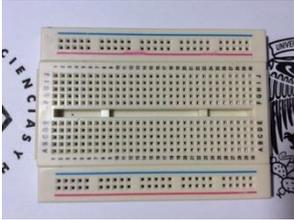
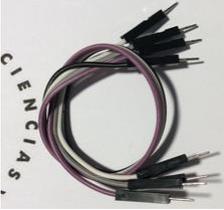


Figura1. A la izquierda se muestra la forma tradicional de medir temperatura con termómetro de mercurio. A la derecha medición con Microcontrolador Arduino y sensores.

MATERIAL

Descripción	Imagen	Comentarios
Microcontrolador Arduino UNO		Hay diferentes tipos de Arduino y el más común es el UNO.
Cable de conexión		Regularmente USB A macho a USB B macho.
Pantalla de Cristal Líquido LCD 16x32 (1602A)		Con led de fondo azul o verde. Hay pantallas sin Led de fondo son más baratas y funcionan casi del mismo modo. Se requiere que el usuario suelde los pines de conexión.
Potenciómetro de 10 K Ω		Resistencia variable para controlar la intensidad de la pantalla LCD. Los extremos van conectados a tierra y la alimentación.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

Descripción	Imagen	Comentarios
Protoboard		Existen de diferentes tamaños. Uno pequeño está bien para estos proyectos.
Jumpers macho-macho		Se necesitan varios, se recomienda adquirir por lote de 40 cables.
Jumpers macho-hembra		Se necesitan varios, se recomienda adquirir por lote de 40 cables.
Sensor digital de temperatura DS18B20		Se recomienda poner un poco de soldadura en los cables del sensor para poder conectar en el protoboard o en los Jumpers hembra.
Resistencia 4.7 KΩ a medio watt.		Se suelen vender por paquete de 5.
Computadora con programa Arduino.		Se recomienda tener instalada la última versión estable de Arduino.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

Descripción	Imagen	Comentarios
Batería de respaldo de 2000 mAh, o capacidad similar		Existen diversas marcas. Revisar que estén cargadas.

Para el desarrollo de prácticas en esta estrategia se requiere además el siguiente material:

- Una parrilla (eléctrica o convencional)
- Un vaso de precipitado de 100 ml
- Un soporte universal
- Una pinza para termómetro
- Sustancias líquidas no inflamables a temperaturas menores de 100 °C (agua, aceite, etc.)



Figura 2. Estudiantes montando el diseño experimental para comparar medición de temperatura de calentamiento y enfriamiento para dos líquidos (agua y aceite).

Software

- Programa Arduino IDE 1.8.2 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>.
- El manejo de datos se puede realizar en hojas de datos o programas para graficar (Excel, Graph, Numbers o Datasheet).
- Programa Fritzing 0.9.3b o superior. Disponible en <http://fritzing.org/download/>. Es recomendable para que los alumnos manejen las conexiones de los sensores de forma virtual.
- Existen algunas bibliotecas especiales para algunos sensores, por ejemplo, el ds18b20 requiere las bibliotecas wire.h y dallastemperature.h. Para su instalación se puede consultar el Tutorial de Arduino (Bañuelos, 2017).

DESARROLLO

En esta práctica se debe tomar en cuenta previamente las siguientes consideraciones:

- Armar y programar las placas de Arduino con los sensores indicados.
- Cargar el programa que será manejado por la placa (el microcontrolador, Arduino UNO).

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

- Asegurarse que las conexiones sean correctas, sobre todo en los sensores pues se corre el riesgo de quemaduras, cortos circuitos y daño al equipo.
- Contar con un equipo de cómputo funcional para programar las instrucciones al microcontrolador.
- Tomar precauciones con el manejo de las parrillas y sustancias calientes.
- El sensor LM35 **No** es de inmersión, por lo que debe asegurarse que solo se utiliza en contacto exterior a recipientes o superficies. Nunca sumergirlo en líquidos.
- Calibrar el sensor, aunque las especificaciones sugieren que ya está calibrado. Siempre es mejor comparar las mediciones con otro sensor o instrumento de medición ya calibrado.

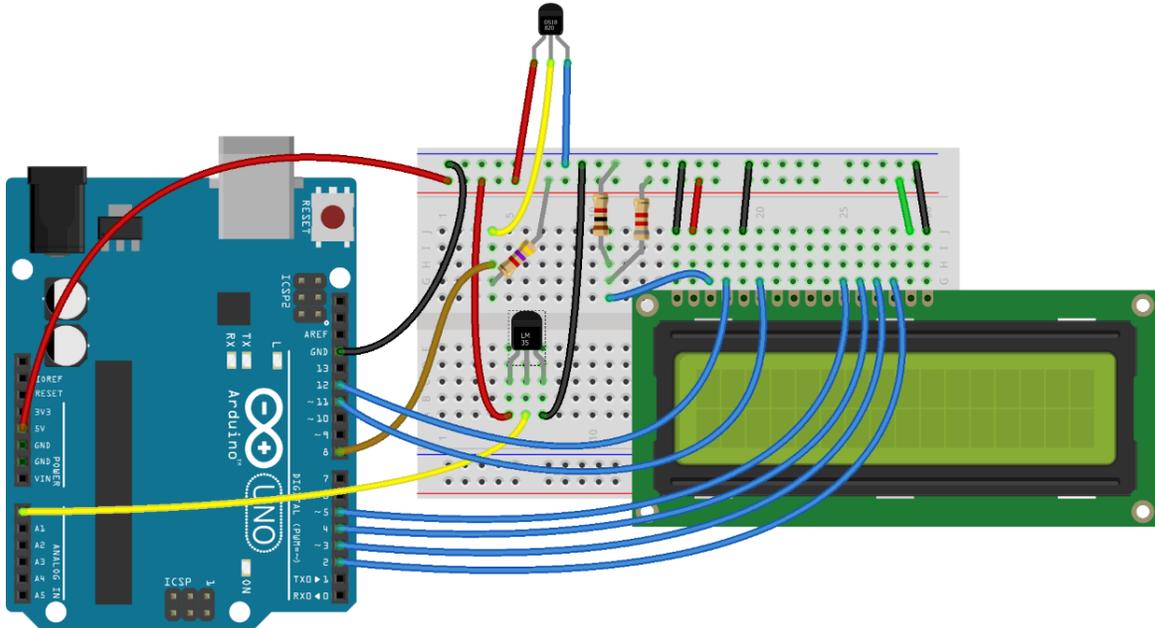


Figura 4. Esquema de conexión del microcontrolador, pantalla LCD y los sensores de temperatura. Observe que el sensor LM35 (al centro) se conecta al pin A0 del Arduino, mientras que el sensor DS18B20 de inmersión (parte superior) va conectado con una resistencia de 4.7k. En la figura no aparece el sensor en su versión con encapsulado metálico.

La calibración de un instrumento es una parte esencial del proceso de medición y es muy importante para fomentar la cultura de las personas pues a menudo ocurre que desconfiamos o confiamos mucho de los instrumentos de medición de artículos y consumos de primera necesidad (combustibles, alimentos, energía eléctrica, etc.).

La calibración en un sensor resulta un poco complicada si no tenemos un instrumento ya calibrado. Afortunadamente en el colegio contamos con algunos termómetros de mercurio, por lo que podemos comparar el grado de confiabilidad de los sensores.

Resulta que las especificaciones de los sensores que utilizamos dan una precisión de hasta medio grado centígrado de temperatura bajo ciertas condiciones. Al poner en funcionamiento dos sensores resulta poco probable que registren la misma temperatura y que estos se encuentren descalibrados, bajo esta premisa podemos trabajar con cierta garantía de que los datos que obtendremos son confiables.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

Una vez revisadas las consideraciones se recomienda consultar el tema 6 de Física en la Enciclopedia de Conocimientos Fundamentales² (UNAM, 2014). Es importante destacar que en un termómetro de mercurio se observa de forma directa el cambio en la longitud de la columna de este líquido metálico, como consecuencia directa de su exposición a cambios de temperatura. Con el uso de sensores podemos decir que ocurre algo similar, pero cambian otras propiedades del material que no pueden ser vistas de forma tan directa (la resistencia o el voltaje, por ejemplo).

Actividad 1. Caracteriza tu instrumento de medición.

Investiga y resume cómo funcionan los sensores que empleas para tus mediciones (Mathas, 2011) (Maxim Integrated, 2018), y (Quora, 2014). Llena la siguiente tabla.

Instrumento	Rango de operación	Resolución	Observación
LM35			
DS18B20			
Hg			
Otro			

Para proceder con la práctica pon 100 ml de agua en el vaso de precipitado.

1. Coloque el vaso con el contenido sobre la parrilla.
2. Ponga un soporte Universal con pinza para el sensor DS18B20 y un termómetro de Hg.
3. Sujete el sensor con la pinza e introdúzcalo en el contenido del vaso sin tocar las paredes.
4. Asegúrese de que la lectura del sensor no sea muy superficial ni en el fondo del vaso.
5. Coloque el sensor LM35 en contacto con el exterior del vaso.
6. Encienda la parrilla y tome lectura del calentamiento del líquido cada 30 s.
7. Cuando observe vapor y burbujas del líquido apague la parrilla si la temperatura ya no aumenta (considere que ya no aumenta cuando en varias tomas de lectura se repite aproximadamente el mismo valor registrado, recuerde que tomamos una precisión de medición de más o menos medio grado centígrado).
8. Para la lectura de datos en el enfriamiento, cambie con cuidado la base de la parrilla por una base de madera o de material aislante o conductor de calor, lo cual modificará sus lecturas. Podría medir para ambos casos la diferencia en enfriamiento.
9. Inmediatamente siga con la lectura de datos de la temperatura ahora cada 60 s en la fase de enfriamiento.
10. Considere aproximadamente un tiempo de 10 min. para calentamiento y 30 para enfriamiento.

² Editada por UNAM-Siglo XXI, páginas 405-425. Se desarrolla brevemente el concepto de temperatura.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

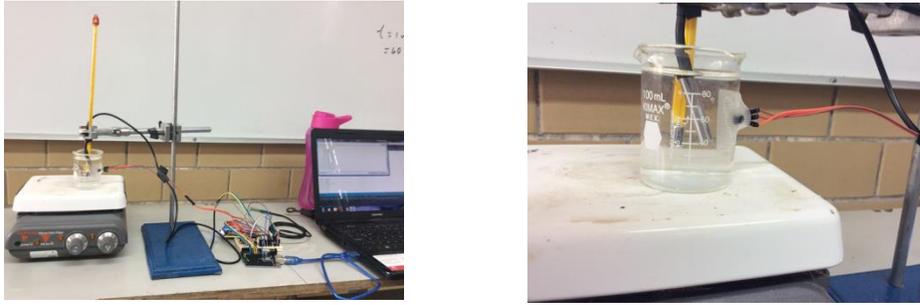


Figura 5. Diseño experimental. Obsérvese la colocación de los sensores y el seguimiento de los datos en tiempo real. Las versiones recientes de Arduino permiten un seguimiento gráfico al momento de la captura de datos de temperatura vs el tiempo.

Actividad 2. Registre sus datos y ajuste un modelo para describir el comportamiento.

Recuerde siempre registrar los valores iniciales de la temperatura. Para ello deje transcurrir un tiempo en contacto los termómetros con el objeto en estudio. La lectura de datos se puede observar en la pantalla LCD del dispositivo o si se cuenta con computadora, a través del puerto serial se pueden tomar los datos. En versiones modernas del Arduino se pueden hacer gráficas en tiempo real.



Figura 6. Registro de lecturas de temperatura en la pantalla de cristal líquido.

- I) Registre los valores iniciales de la temperatura. Los registros deben incluir la fecha, la hora y el lugar donde se realizan las mediciones. Utiliza la siguiente tabla o diseña la tuya propia para dejar nota de los resultados.

Termómetro	Temperatura inicial	Observación
LM35		
DS18B20		
Hg		

- II) Registre las lecturas cada 30 segundos para el calentamiento y después cada 2 minutos para el enfriamiento si se encuentra observando en la pantalla de cristal líquido.

Tiempo (s)	T(°C) LM35	T(°C) DS18B20	T(°C) Hg

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

Para la captura de datos en la computadora asegúrese de configurar el puerto serial para comunicar la computadora con el Arduino. Seleccione los datos con el ratón y cópielos a un bloc de notas. Después abra el archivo con una hoja de cálculo para realizar las gráficas y los ajustes.

- III) A continuación, realice un gráfico en dispersión para las lecturas de la temperatura contra el tiempo.
- IV) Finalmente, analice los comportamientos de los resultados y redacte un informe breve.

No se olvide de obtener un modelo matemático para cada termómetro y comparar para el caso del calentamiento las pendientes de los modelos ajustados si es que corresponde.

$$T = mt + b$$

Para las curvas obtenidas en el enfriamiento intente un ajuste con un modelo exponencial decreciente.

$$T = T_0 e^{-at} + B$$

Actividad 3. Conteste las siguientes preguntas y justifica tus respuestas.

1. ¿Por qué es necesario esperar un tiempo con los termómetros en contacto con el sistema antes de tomar lecturas?
2. ¿Cuál de los termómetros es más confiable? ¿Por qué?
3. ¿A qué se debe que la relación entre temperatura y tiempo se pueda aproximar con una relación lineal para el calentamiento? ¿Hay algunas restricciones?
4. ¿Qué representa la pendiente de los modelos ajustados?
5. ¿Cómo sería la curva de calentamiento (las pendientes) para sustancias diferentes por ejemplo aceite o atole?
6. ¿Depende el calentamiento de la densidad de las sustancias?
7. ¿Por qué razón la curva de enfriamiento es diferente de la curva de calentamiento?
8. ¿Se enfría la sustancia siempre con la misma rapidez? ¿por qué?
9. ¿Qué representa el exponente a en el modelo de enfriamiento?
10. Redacte unas breves conclusiones.

Actividad 4. Equilibrio térmico

Sabemos que cuando ponemos dos cuerpos en contacto térmico al cabo de cierto tiempo estos alcanzan el equilibrio en su intercambio neto de energía, por lo que ambos alcanzan el mismo valor en la energía cinética promedio de sus partículas. Es difícil observar el comportamiento a nivel microscópico de las partículas, pero asociamos ese valor promedio de la energía cinética con la temperatura que se manifiesta en el cambio de una variable macroscópica medible.

Para realizar esta actividad pondremos en contacto térmico tres sistemas para ver su evolución de la temperatura en el tiempo y corroborar que se alcanza un equilibrio manifestado por el mismo valor de la temperatura en los tres sistemas.

Se recomienda consultar información sobre equilibrio térmico y ley cero de la termodinámica. Para una consulta rápida se la monografía de Olmo y Nave.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

Lleve a cabo los siguientes pasos:

1. Caliente 100 ml de agua en un vaso pequeño de precipitado hasta una temperatura cercana a los 90 °C
2. Enfríe 100ml de agua en un vaso pequeño de precipitado hasta una temperatura cercana a 0 °C
3. Vacíe 1 litro de agua a temperatura ambiente en un recipiente amplio.
4. Proponga un valor final de la temperatura al cabo de 10 minutos o más.
5. Introduzca los vasos de precipitado en el recipiente y determine la evolución temporal de la temperatura con los sensores DS18B20. ¿Se cumple tu predicción? ¿Se podría predecir la temperatura final del sistema?
6. ¿Qué sucede con la energía de cada uno de los sistemas antes y después de ponerlos en contacto?

Actividad 5. Temperatura en líquidos diferentes

Repita los procedimientos de las actividades anteriores, solo que ahora compare el calentamiento y el enfriamiento de dos sustancias al mismo tiempo.

- I) Realice algunas predicciones de acuerdo con el tipo de sustancia. Puede estudiar el calentamiento de acuerdo con la cantidad de sustancias, la densidad, el recipiente en que se calienta, la base en la que se enfría, etc.
- II) Compare sus gráficas y analízaslas.
- III) Obtenga una conclusión de lo que se observó.

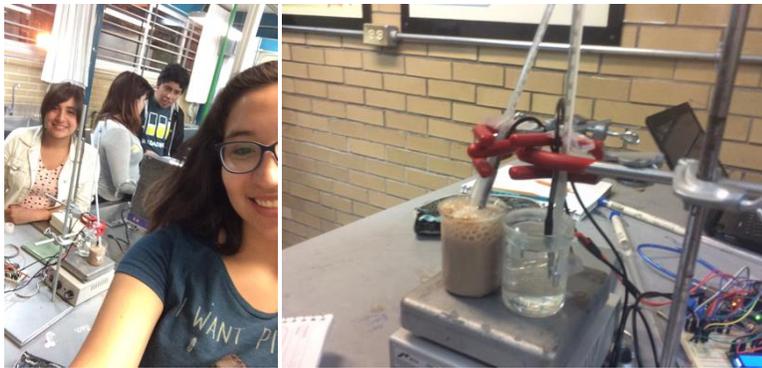


Figura 7. Alumnos llevando a cabo la comparación de calentamiento entre dos líquidos diferentes.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

¿Qué otra situación o fenómeno físico se le ocurre medir o estudiar con este dispositivo?

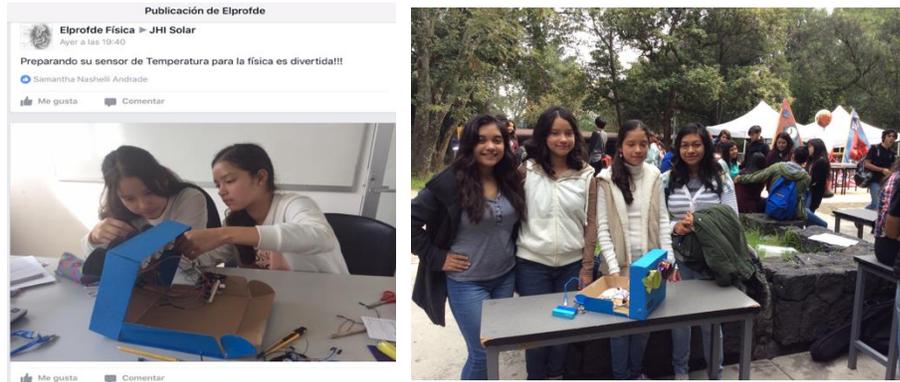


Figura 8. Estudiantes armando un dispositivo para medir temperatura de la radiación Solar a través de filtros de color. Implementaron tres filtros rojo, azul y verde.

RESULTADOS

(Resultados y análisis de resultados: sugerencias de resultados, gráficas probables y formas de analizar)

Las especificaciones de los sensores de temperatura señalan una precisión de ± 0.5 °C. Los termómetros de mercurio tienen como marcas mínimas de precisión 1°C. El sensor LM35 es sensible a los cambios de temperatura, pero solo mide en interiores al contacto, además tiene la desventaja de no ser de inmersión. Su intervalo de operación va de los -50 °C a los 150 °C. Al medir presenta ruido electrónico lo cual se traduce en fluctuaciones al graficar los valores, ver figura 9.

Los sensores DS18B20 registran menores efectos de fluctuación en el valor de la temperatura. Su operación requiere bibliotecas que se deben instalar de forma particular. Tienen especificaciones similares a las del LM35 pero con la ventaja que el DS18B20 sí es un sensor de inmersión. Se recomiendan estos sensores por su versatilidad.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

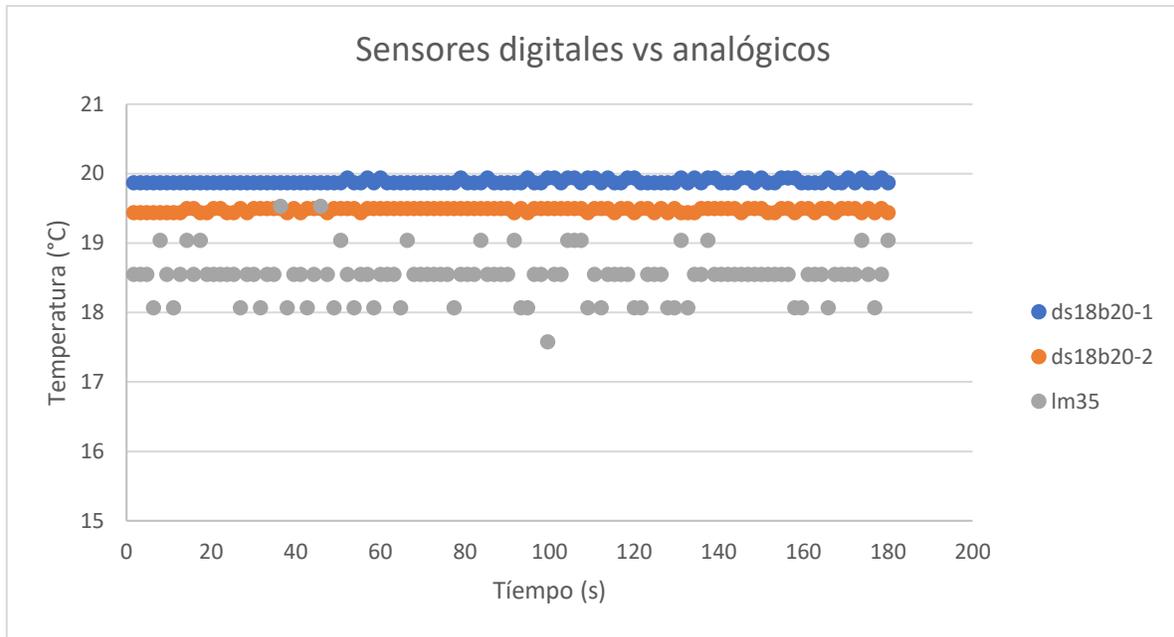


Figura 9. Lecturas de la temperatura ambiente con los sensores analógico LM35 y digitales ds18b20. Se puede observar fuertes fluctuaciones en el sensor analógico y más regularidad en los digitales. Hay diferencias significativas entre los sensores de 0.5 a 1.5 °C.

En gráficos de estudio para el equilibrio térmico se esperan resultados como los que se muestran en la siguiente figura 10. Cabe mencionar que en realidad aquí la temperatura de equilibrio corresponde a tres termómetros puestos en contacto con los tres cuerpos. Puede decirse que uno de los cuerpos está en contacto térmico con los otros dos, ya que uno de ellos contiene a los otros. Se observa la convergencia de la temperatura de los tres cuerpos a un valor.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

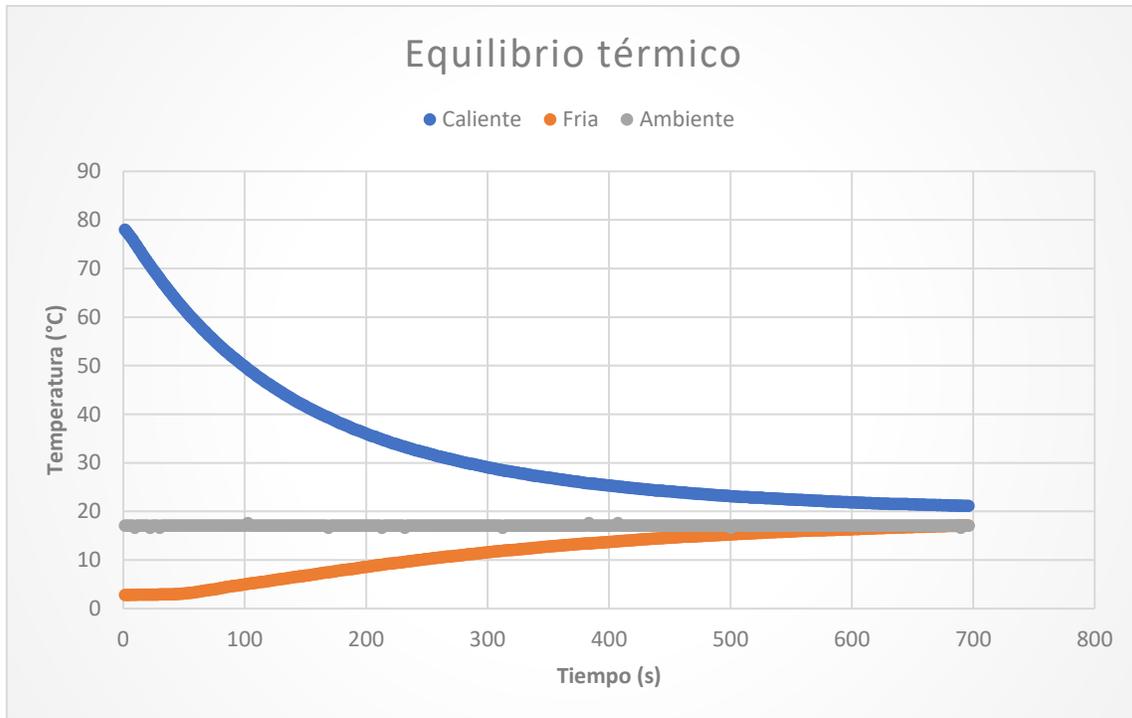


Figura 10. Gráfico de la temperatura de tres cuerpos puestos en contacto térmico. En azul está la gráfica de 100 ml de agua que se enfría desde 80 °C, en gris hay un litro de agua puesto a temperatura ambiente 19 °C y en rojo se representa 100 ml de agua a 3 °C. Se aprecia que la temperatura final de los tres recipientes con agua será la del de mayor tamaño.

El calentamiento y enfriamiento de sustancias puede resultar en gráficas como la que se muestran a continuación. En este ejemplo utilizamos agua simple. Se sugiere comparar con mezclas para probar la capacidad de predicción de los alumnos. Obsérvese que el comportamiento inicial es no lineal, pero después de algún par de centenas de segundos ya se comporta aproximadamente lineal.

El agua en el Colegio hierve a temperaturas que oscilan entre los 92 y 94 °C dependiendo de las condiciones ambientales. Es importante señalar que los estudiantes consideran que el agua simple hierve hasta los 100 °C, por lo que les comentan en la secundaria o por lo que ven en los libros. Este ejercicio aclara que las condiciones no son las del nivel del mar, o al menos que son diferentes.

Resulta conveniente dividir los datos en dos gráficas, una para calentamiento y otra para enfriamiento pues los análisis son diferentes. Nótese que en este ejercicio es conveniente tomar lecturas durante aproximadamente el mismo intervalo de tiempo para calentamiento que para enfriamiento.

El enfriamiento tiene un comportamiento que se puede estimar bien a través de un ajuste exponencial decreciente. Es pertinente que después de la práctica el alumno consulte la ley de enfriamiento de Newton.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

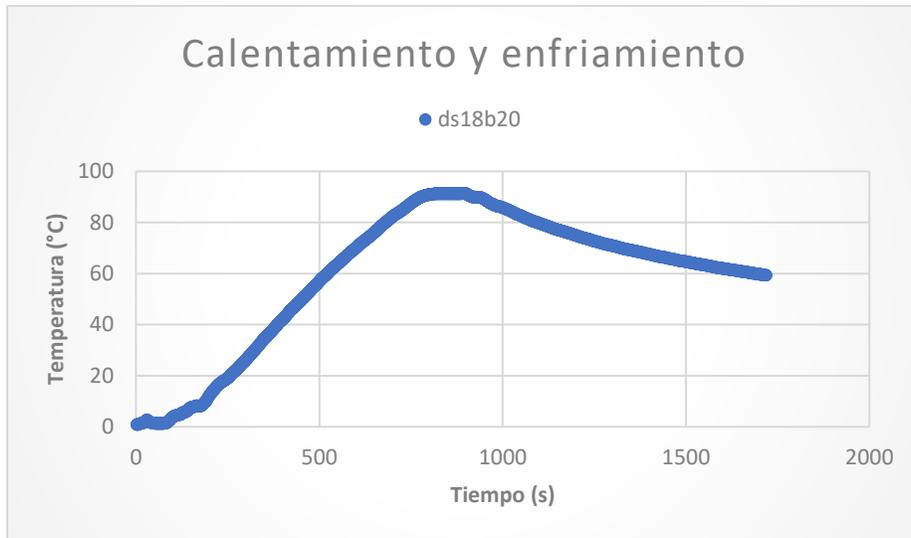


Figura 11. Curva de calentamiento y enfriamiento de 100 ml de agua. Los datos fueron registrados con el sensor DS18B20, se usó una parrilla de gas. Bajo un suministro aproximadamente constante de energía calorífica a los 100 ml de agua se encuentra un aumento de temperatura medido por el sensor que puede aproximarse a una relación lineal con el tiempo, mientras que para el enfriamiento se observa una relación decreciente.

En el enfriamiento se pueden obtener ajustes con exponencial decreciente.

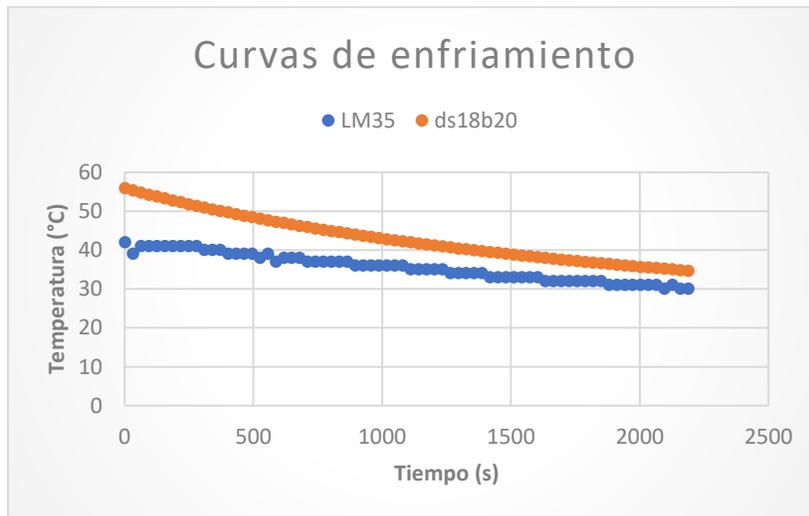


Figura 12. Curvas de enfriamiento con sensores LM35 y DS18B20. Enfriamiento de 100 ml de agua, el sensor LM35 fue colocado en contacto con el vaso de precipitado y el sensor DS18B20 al interior del líquido.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

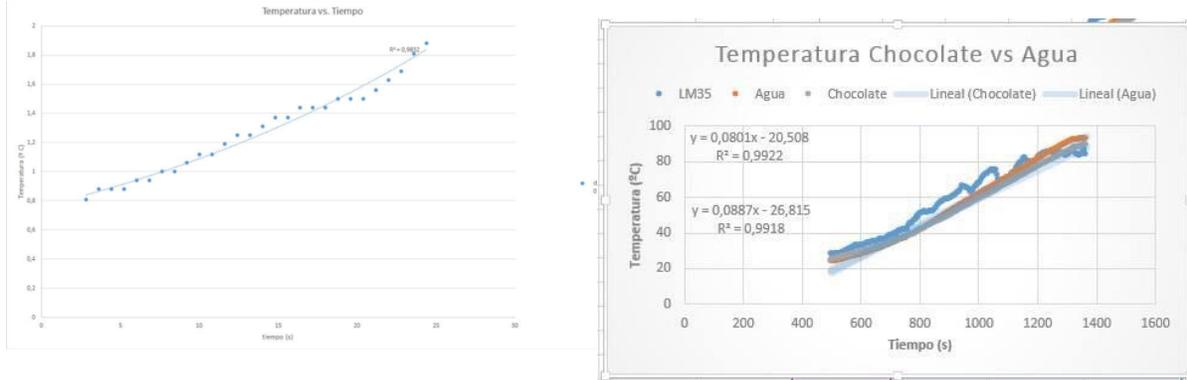


Figura 13. Ajuste del calentamiento realizado por estudiantes. Se probaron varios ajustes y resultó una mejor correlación polinomial de grado 2 para pocos puntos al inicio del calentamiento de agua (a la izquierda). Ajuste lineal al calentamiento de agua y chocolate con leche, propuesta de alumnos para saber que se calienta más rápido (derecha).

DISCUSIÓN Y CONCLUSIONES

El uso de microcontrolador Arduino con sensores para medir la temperatura resultó ser un desafío para los estudiantes. Comprendieron que el uso de algunas tecnologías contribuye a mejorar la calidad de los datos obtenidos en prácticas experimentales y que si bien, como en todo nuevo conocimiento, la curva de aprendizaje al principio puede ser lenta, con el manejo frecuente se aprende y se mejoran las habilidades, se consolida el aprendizaje.

Se aprendió la aplicación de sensores para la medición de temperatura, así como las ventajas y desventajas de cada sensor. Por ejemplo, comprendimos que el sensor LM35 es un sensor barato y sensible para la medición de temperatura en interiores y solo de contacto pero no de inmersión. Por otra parte, el sensor DS18B20 resultó más interesante, pues además de tener más estabilidad en las lecturas es un sensor con más versatilidad, ya que permite inmersión en líquidos.

La aplicación de sensores eléctricos fomentó en los alumnos el interés en su aplicación y en entender su funcionamiento. En las prácticas experimentales de física sería ideal implementarse el uso de estos equipos de bajo costo para desarrollar estrategias didácticas que tengan una formación integral en las nuevas generaciones de alumnos.

Consideramos que el alumno debe alcanzar la habilidad de montar una placa Arduino con un sensor de temperatura y realizar algunas de las actividades sugeridas aquí. Se debe hacer la invitación a una propuesta de aplicación del sensor para responder con datos duros una pregunta planteada por los alumnos mismos esto promueve la actitud científica y la innovación en la aplicación tecnológica y de conocimiento.

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

REFERENCIAS

Applets de la Universidad de Colorado sobre calor y temperatura. Recuperado el 5 de junio de 2019 de <https://phet.colorado.edu/es/simulations/category/physics/heat-and-thermodynamics>

Bañuelos, M. A. (2017). Tutorial de Arduino. México: ICAT-UNAM.

Bibliotecas especiales para el sensor DS18B20. Recuperado el 5 de junio de 2019 de <https://github.com/milesburton/Arduino-Temperature-Control-Library>

Bueche, F. J. (1991) Fundamentos de física. Tomo I. México: McGraw Hill.

CCH UNAM(2003) Plan de estudios física I a IV. Recuperado el 5 de junio de 2019 de http://www.cch.unam.mx/sites/default/files/plan_estudio/mapa_fisica.pdf

Martínez, M., (2012). Termodinámica. México: UNAM.

Mathas, C. (2011). Conceptos Básicos sobre sensores de temperatura. Recuperado el 5 de junio de 2019 de <http://www.digikey.com/es/articles/techzone/2011/oct/temperature-sensors-the-basics>

Maxim Integrated (2018). Hoja de datos del sensor digital de temperatura ds18B20. Recuperado el 5 de junio de 2019 de <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

Olmo, M. y Nave, R. (). ¿Qué es la Temperatura?. Recuperado el 5 de junio de 2019 de <http://hyperphysics.phy-astr.gsu.edu/hbasees/thermo/temper.html>

Quora (2014). How-does-LM35-work. Recuperado el 5 de junio de 2019 de <https://www.quora.com/How-does-LM35-work>

Resnick, R., Halliday, D. y Krane, K. (2001) Física Vol. 1. México: CECSA.

UNAM (2014). Enciclopedia conocimientos fundamentales UNAM Siglo XXI. Recuperado el 5 de junio de 2019 de <http://www.unamenlinea.unam.mx/recurso/enciclopedia-de-conocimientos-fundamentales-unam-siglo-xxi-matematicas>

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

ANEXOS

Programa 1 con cuatro termómetros simultáneos tres digitales ds18b20 y un LM35

```
/*
*****
* Autor: Sergio Alejandro Carrillo Araujo *
* CCH Sur UNAM *
* CC:3.0 *
* 16/agosto/2015 Última revisión: 10/09/2016 *
* Proyecto: Termómetro Arduino *
* Estado: Funcional *
* Material: Arduino Leonardo, UNO, Display 1602A, *
* Sensores ds18b20, LM35, Potenciómetro 10K *
* Propósito: Medir temperatura *
* *
* Detalles: Hay que calibrar el sensor *
*****
*/
```

```
// Para display LCD incluir librería
#include <LiquidCrystal.h>
```

```
//Libreria I2C (para sensor digital de temperatura ds18b20)
#include<DallasTemperature.h>
#include <OneWire.h>
```

```
/*
*****
Seguir las conexiones del display LCD al Arduino
* VSS a GND
* VDD a 5 Volts
* VO a GND
* RS a pin digital 6
* R/W a GND
* E a pin digital 7
* DB4 a pin digital 5
* DB5 a pin digital 4
* DB6 a pin digital 3
* DB7 a pin digital 2
*****
* Sensor LM35
* Especificaciones: Cada grado Celsius equivale a 10 mV
* Rango: -55 °C a 150 °C
* Precisión: 0.5 °C
* Corriente: 60 microA
* (visto con cara plana de frente)
* Iq=Vs a 5 V
* Cent=Vout a salida A0
* Der= GND
*****
*/
```

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

```
* Potenciómetro
* Izq a GND
* Cent a Vo
* Der a 5V
* *****
*/

/*Integrar nuevos objetos*/
//Pin análogo para el sensor
const int tempPin =0;
const int tempPin1 = 1;
const int tempPin2 = 2;
const int tempPin3 = 3;
//Pin a donde está conectado el ánodo de los leds de iluminación de la pantalla
int brightnessPin = 10;
// Variable de la temperatura
double tempC,tempC1, tempC2, tempC3, tempF, tempK;
// para contar los segundos
int ss = 0;
int mm = 0;
int hh = 0;

// para sensor d
#define wire 8
// Activa controlador de protocolo
OneWire oneWire(wire);
// Asigna controlador de protocolo y nombre de sensor
DallasTemperature sensors(&oneWire);

//Declarar pines usados por pantalla
LiquidCrystal lcd(7,6,5,4,3,2);

void setup() {

//iniciamos el sensor de temperatura
sensors.begin();

// Para ver el puerto serial
Serial.begin(9600);
Serial.println("Temp vs Time");

// Configuración de los pines
// Leds al 100%
// analogWrite(brightnessPin, 60);
// Tamaño de pantalla
lcd.begin(16,2);
// Cursor en la primera línea
lcd.setCursor(0,0);
```

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

```
// Escribir primer mensaje
lcd.print("Buen dia Alex");
delay(2000);
//borrar pantalla
lcd.clear();

}
void loop()
{
  //Función para leer datos del sensor
  tempC=analogRead(tempPin);
  tempC1=analogRead(tempPin1);
  tempC2=analogRead(tempPin2);
  tempC3=analogRead(tempPin3);
  //Mantener la luz de la pantalla encendida
  // digitalWrite(brightnessPin,HIGH);
  {
    //Conversión de temperaturas
    //Celsius
    tempC=(5.0*tempC*100.0)/1023.0;
    tempC1=(5.0*tempC1*100.0)/1023.0;
    tempC2=(5.0*tempC2*100.0)/1023.0;
    tempC3=(5.0*tempC3*100.0)/1023.0;
    //Kelvin
    tempK=(tempC+273.15);
    //Fahrenheit
    tempF=(tempC*9)/5+32;
  }
  lcd.setCursor(0,0);
  lcd.print("Temp: ");
  lcd.print(tempC);
  lcd.print((char)B11011111);
  lcd.print("C");

  // una alerta de temperatura!!
  if(tempC>=98){

    analogWrite(0,LOW);
    delay(1000);
    analogWrite(0,HIGH);

    lcd.print(tempC);
    lcd.print((char)B11011111);

    lcd.print("CCH");
```

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

```
        lcd.setCursor(0,1);
        lcd.print(tempF);
        lcd.print((char)B11011111);
        lcd.print("F ");
        lcd.print("CCH SUR");
        lcd.clear();

    }
    // Si no se cumple la condición anterior entonces lo siguiente
    else{
        lcd.setCursor(5,0);
        lcd.print(tempC);
        lcd.print((char)B11011111);
        lcd.print("C CCH");
        lcd.setCursor(0,1);
        lcd.print(tempF);
        lcd.print((char)B11011111);
        lcd.print("F ");
        lcd.print("UNAM SUR");
    }

    /*****/
    sensors.requestTemperatures();
    //tiempo
    Serial.print(millis()/1000.0);

    Serial.print(" ");

    Serial.print(tempC);
    Serial.print(" ");
    Serial.print(tempC1);
    Serial.print(" ");
    Serial.print(tempC2);
    Serial.print(" ");
    Serial.print(tempC3);
    Serial.print(" ");

    Serial.println(sensors.getTempCByIndex(0));

}
    /*****FIN*****/
```

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

Programa 2 con tres termómetros simultáneos dos de lectura digital ds18B20 y un LM35.

```
/*
*****
* Autores: Edith Rojas, Natalia Soto, Iván Olvera. *
* Sergio Alejandro Carrillo Araujo *
* CCH Sur UNAM *
* CC:3.0 *
* 16/agosto/2015 Última revisión: 10/09/2016 *
* Proyecto: Termómetros con Arduino *
* Estado: Funcional *
* Material: Arduino Leonardo, UNO, Display 1602A, *
* 2Sensores ds18b20, LM35, Potenciómetro 10K *
* Propósito: Comparar temperatura *
* *
* Detalles: Podemos comparar distintas sustancias *
*****
*/
```

```
#include <DallasTemperature.h>
#include <OneWire.h>
#include <LiquidCrystal.h>

OneWire wire(8);
DallasTemperature sensor1(&wire);
OneWire wire2(9);
DallasTemperature sensor2(&wire2);
LiquidCrystal lcd(7,6,5,4,3,2);
const int tempPin=0;
float temp0,temp1,temp2,tempC,tempF,tempK;

void setup() {

  Serial.begin(9600);
  Serial.println("Tiempo vs Temp");

  sensor1.begin();
  sensor2.begin();
  lcd.begin(16,2);
  lcd.print("Hola Edith");
  delay(3000);
  lcd.clear();
}

void loop() {
  sensor1.requestTemperatures();
  temp1=sensor1.getTempCByIndex(0);
  sensor2.requestTemperatures();
```

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

```
temp2=sensor2.getTempCByIndex(0);
temp0=analogRead(tempPin);
tempC=(500/1024.0)*temp0;
tempF=(tempC*9)/5+32;
Serial.print(millis()/1000.0);
Serial.print(" ");
Serial.print(tempC);
Serial.print(" ");
Serial.println(temp1);
Serial.print(" ");
Serial.println(temp2);
```

```
lcd.setCursor(0,0);
lcd.print("Temp= ");
lcd.print(tempC);
lcd.print((char)B11011111);
lcd.print(" CCH");
lcd.setCursor(0,1);
lcd.print(temp1);
lcd.print(" ");
lcd.print(temp2);
delay(2500);
}
/*****FIN*****/
```

Programa 3 Básico con un sensor d18b20 y un LM35.

```
#include <DallasTemperature.h>
#include <OneWire.h>
#include <LiquidCrystal.h>
OneWire wire(8);
DallasTemperature sensor1(&wire);
LiquidCrystal lcd(7,6,5,4,3,2);
const int tempPin=0;
float temp0,temp1,tempC,tempF,tempK;
void setup() {
  Serial.begin(9600);
  Serial.println("Tiempo vs Temp");
  sensor1.begin();
  lcd.begin(16,2);
  lcd.print("Hola Edith");
  delay(3000);
  lcd.clear();
}
void loop() {
```

7. Curvas de calentamiento y enfriamiento de líquidos con sensores en Arduino

```
sensor1.requestTemperatures();
temp1=sensor1.getTempCByIndex(0);
temp0=analogRead(tempPin);
tempC=(500/1024.0)*temp0;
tempF=(tempC*9)/5+32;
Serial.print(millis()/1000.0);
Serial.print(", ");
Serial.print(tempC);
Serial.print(", ");
Serial.println(tempF);
lcd.setCursor(0,1);
lcd.print("Temp= ");
lcd.print(tempC);
lcd.print((char)B11011111);
lcd.print("C");
lcd.setCursor(0,1);
lcd.print(temp1);
delay(500);
}
```

8. MEDICIÓN DE DISTANCIA CON MOVIMIENTO ROTACIONAL

Responsable: Javier de Jesús Fonseca Madrigal CCH-Sur	Participantes:	Revisión: Eduardo José Vega Murguía
---	----------------	--

Ubicación curricular

Colegio de Ciencias y Humanidades. Programa de estudios versión 2016

Física I

Unidad 2. Movimiento Rectilíneo Uniforme

Física III

Unidad 1. Sistemas de cuerpos rígidos

INTRODUCCIÓN

En la enseñanza experimental de la mecánica es común que se presente la necesidad de contar con un objeto que se mueva a rapidez constante, y además que lo haga despacio para que los alumnos puedan tomar mediciones. Además, es recomendable utilizar objetos que motiven y llamen la atención del alumno, haciendo así agradable la clase para él. En este contexto se diseñó un carrito eléctrico que arrastra a una rueda que gira conforme el carro avanza, y que cuenta con un sensor de movimiento rotacional que permite al Arduino calcular el desplazamiento lineal del carro.

Este dispositivo se diseñó para apoyar los cursos de Física 1, en cuanto a la medición de rapidez constante, y Física 3 en lo que respecta a la relación entre desplazamiento angular y lineal. En este último, con la intención de involucrar a los alumnos en el algoritmo para hacer dichos cálculos, así como en la medición experimental.

El dispositivo, sin el carrito, puede adaptarse para hacer otro tipo de mediciones, como por ejemplo utilizarse como polea para medir movimiento, o como cuenta vueltas. Por el momento sólo se desarrollaron las dos aplicaciones ya mencionadas debido a su aportación directa a la enseñanza aprendizaje, acorde a las necesidades curriculares del programa de estudios de Física 1 y 3.

OBJETIVOS

- El alumno aprenderá a medir fenómenos con rapidez constante mediante el uso de un instrumento didáctico y motivacional (Física 1).
- El alumno aprenderá a medir longitudes utilizando movimiento rotacional (Física 3).

8. Medición de distancia con movimiento rotacional



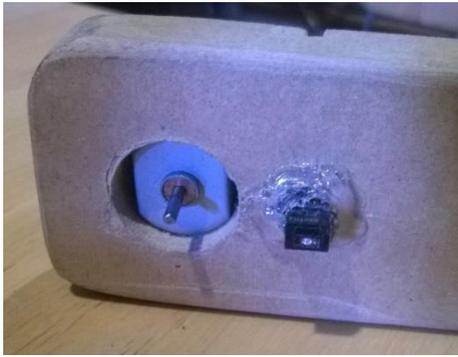
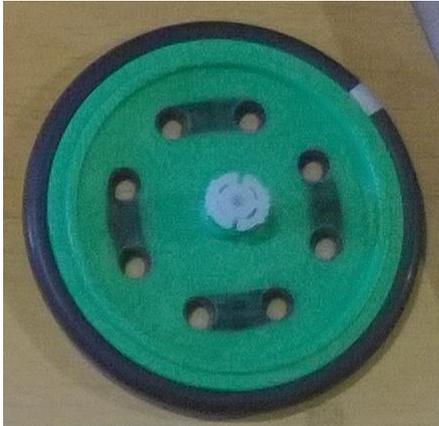
Figura 1. Materiales empleados.

MATERIAL

Descripción	Imagen	Comentarios
<p>Carrito de motor eléctrico con adaptación de madera para soportar brazo con rueda de mediciones</p>		<p>El carrito sin la adaptación de madera se consiguió en Imori Kits³</p>

³ República del Salvador 26 local 9

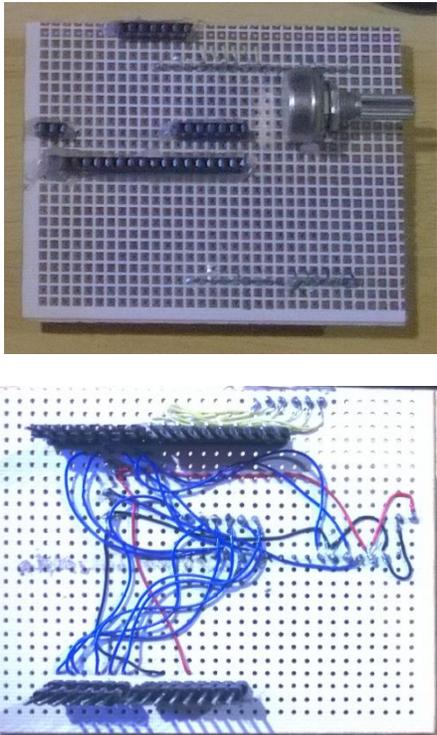
8. Medición de distancia con movimiento rotacional

Descripción	Imagen	Comentarios
Tornillo y mariposa para sujetar brazo con rueda de mediciones al carrito		
Brazo para soportar sensor, eje de rueda y conector tipo teléfono		
Motor eléctrico que servirá como eje para montar la rueda		Se muestra montado.
Rueda con 8 agujeros y adaptador para montar rueda en motor (eje).		<p>El adaptador para acoplar la rueda al eje del motor se extrajo del mecanismo de un motor reductor:</p>  <p>La rueda se consiguió en Imori Kits (ya citado), el motor reductor en "Kinetrónica".</p>

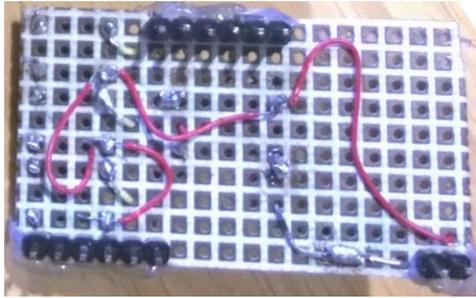
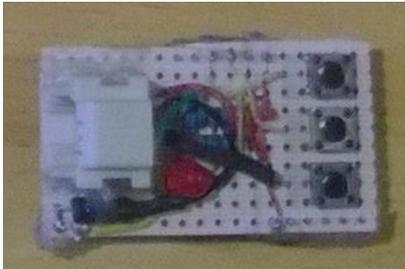
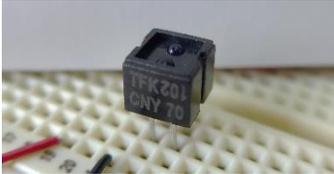
8. Medición de distancia con movimiento rotacional

Descripción	Imagen	Comentarios
2 pilas tamaño AA para hacer funcionar el motor del carrito		
1 batería cuadrada de 9 V con conector para alimentar la corriente del Arduino		
Tarjeta Arduino UNO con protección de acrílico		Se recomienda utilizar un tipo diferente de protector de acrílico, ya que el que se muestra no sujeta con firmeza al Arduino.
Cable USB A macho a B macho		
Cable con conectores para teléfono		Es algo difícil trabajar con estos conectores ya que los cables tienen la polaridad invertida de un conector respecto al otro.

8. Medición de distancia con movimiento rotacional

Descripción	Imagen	Comentarios
<p>Display de cristal líquido con pines para conectarlo</p>		<p>El usuario debe soldar los pines de conexión.</p>
<p>Potenciómetro de 10KΩ</p>		
<p>Tarjeta grande con conectores tipo pin para montar sobre el Arduino UNO</p>		<p>Por ser un prototipo, las tarjetas se soldaron con cables. Este trabajo se podría simplificar si se utilizan tarjetas con circuito impreso.</p> <p>En la parte de arriba se colocaron conectores pin hembra para montar la tarjeta pequeña.</p> <p>En la parte de abajo se tuvo que soldar doble conector porque con el acrílico se requiere mayor distancia para conectar al Arduino.</p> <p>Una de las series de pines que se muestran abajo se tuvo que desplazar hacia un lado porque no coincide la posición en Arduino con la tarjeta.</p>

8. Medición de distancia con movimiento rotacional

Descripción	Imagen	Comentarios
<p>Tarjeta pequeña con conectores tipo pin para insertar sobre la tarjeta grande</p>		<p>Los 3 interruptores que se observan están de más porque se planea añadir algunas funciones en el futuro, como reiniciar el contador, seleccionar tipo de medición (distancia, número de vueltas, rapidez, etc). Véase el diagrama de conexión de las resistencias y el LED en el anexo.</p>
<p>Conector para cable de teléfono 1 LED. 2 resistencias de 220 Ω. 1 resistencia de 20 k Ω</p>		<p>El material negro que se observa es un aislante termoretráctil "Termofit".</p>
<p>1 Sensor infrarrojo QRD1114 o CNY70</p>		<p>Este sensor funciona bien mientras no haya una luz externa que interfiera, como una luz intensa de los focos, o el Sol.</p>
<p>2 conectores hembra para cable de teléfono (RJ11)</p>		<p>La pieza de conector de teléfono originalmente estaba contenida en una cajita para sujetarse a la pared, sin embargo se pudo desensamblar con facilidad.</p>

Software

- Programa Arduino IDE 1.8.2 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>.

8. Medición de distancia con movimiento rotacional

DESARROLLO

Armado del equipo:

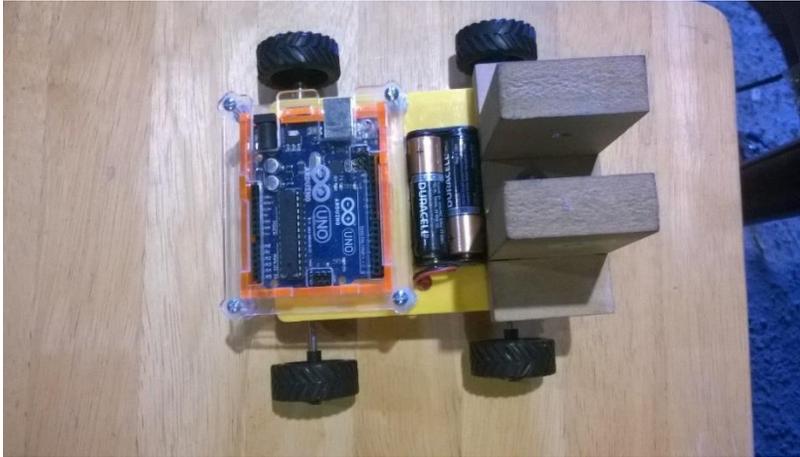


Figura 2. Carro eléctrico con Arduino y baterías montados.

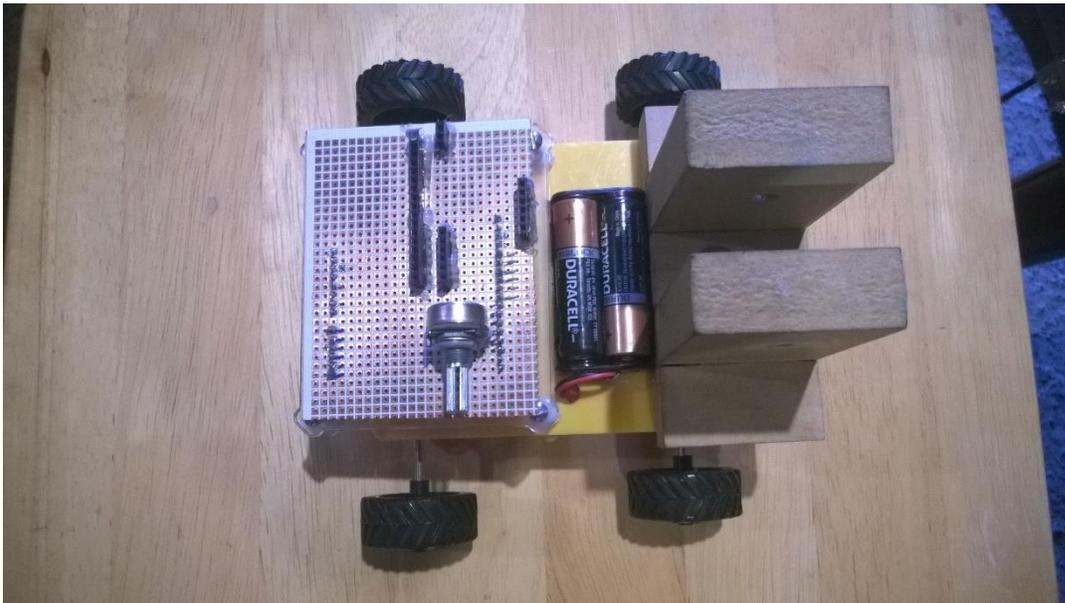


Figura 3. Carro eléctrico con módulo de display montado sobre la tarjeta Arduino.

8. Medición de distancia con movimiento rotacional

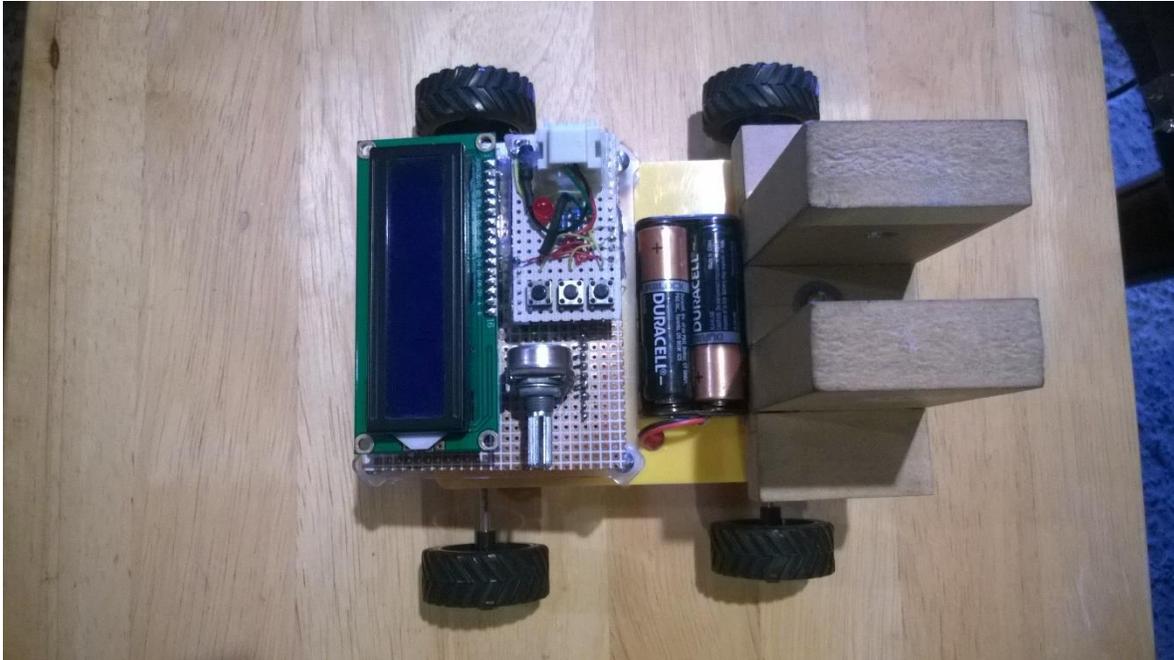


Figura 4. Carro eléctrico con display y módulo de conexiones para sensor montados.



Figura 5. Vista de la rueda montada sobre el brazo de soporte.

8. Medición de distancia con movimiento rotacional

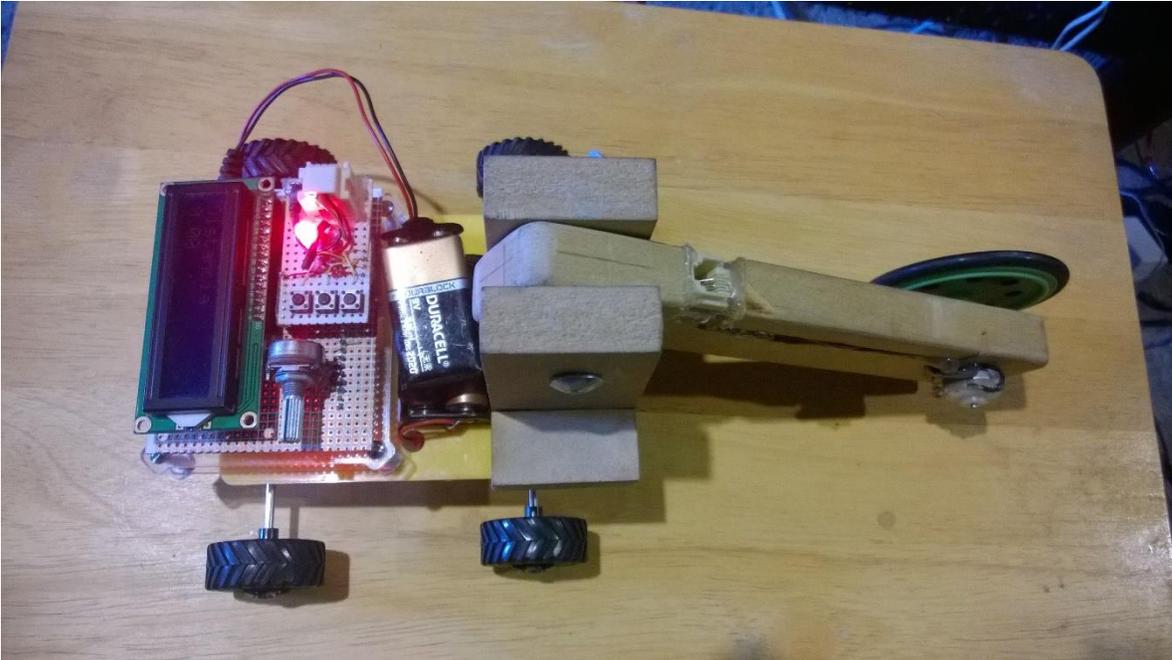


Figura 6. Carrito con batería de 9V para energizar al Arduino.

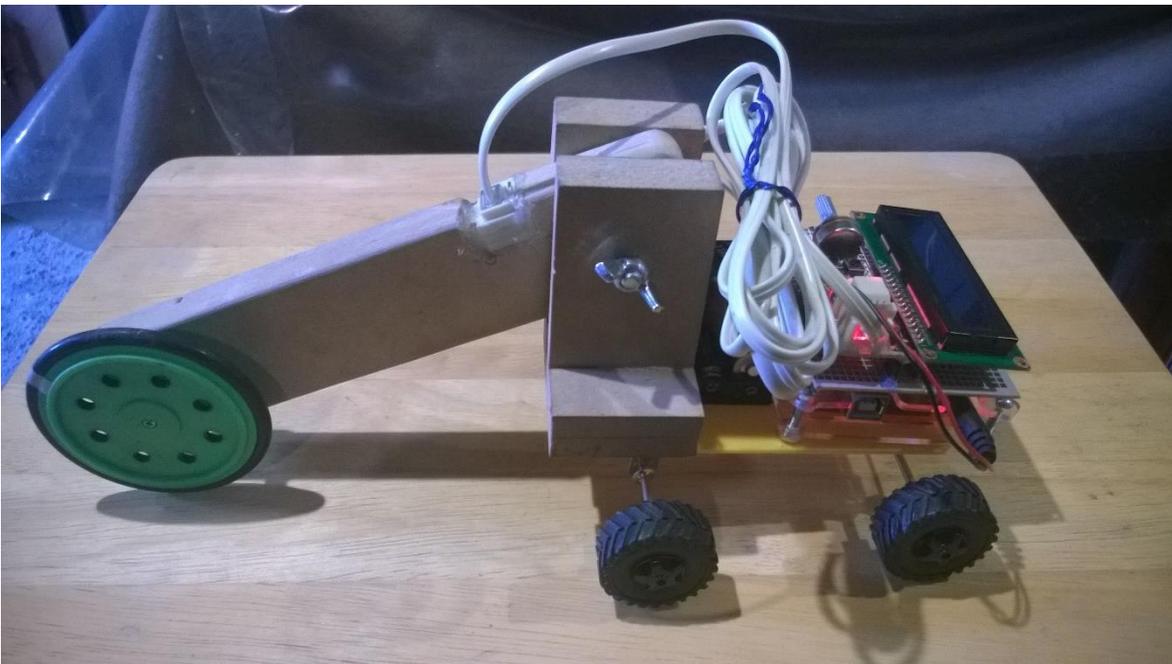


Figura 7. Carrito completamente armado.

8. Medición de distancia con movimiento rotacional



Figura 8. Vista de la lectura del display.

Física 1: Medición de rapidez constante

Para esta medición se requiere contar con un objeto que se mueva con rapidez constante y despacio para que sea medible. Esto se puede lograr cuando de alguna manera se iguala una fuerza impulsora con la resistencia por fricción. El carrito que se presenta cumple con ambas condiciones. Es impulsado por un motor eléctrico a través de una caja de engranes que le permite desplazarse despacio y a velocidad constante.

El experimento consiste en colocar dos marcas de referencia, por ejemplo, con masking tape. Después se hace que el carrito se mueva desde la posición inicial a la final y se mide el tiempo transcurrido. La distancia se puede obtener de la lectura de la pantalla del carrito, con una cinta métrica o ambas. Con estos datos se solicita a los alumnos que calculen la rapidez.

Física 3: Medición de distancia con movimiento rotacional

Se plantea a los alumnos la posibilidad de medir longitudes con una rueda, como la que usan los arquitectos o el odómetro de los coches. Se les pide que tomen como referencia un diámetro de 7.6076 cm (lo que mide el de la rueda sensora del carrito), y encuentren un procedimiento para medir distancias considerando que la rueda tiene marcas para subdividirla en 8 partes iguales.

Después se colocan las marcas de referencia inicial y final, y se procede a hacer la medición con el carrito y con cinta métrica. Los alumnos contarán el número de vueltas que da la rueda medidora del carrito y harán los cálculos correspondientes. Al final se calculan las mediciones de los alumnos con la de la pantalla del carrito y la cinta métrica.

Se concluye que la distancia que recorrió el carrito se conoce como desplazamiento tangencial en movimiento rotacional.

8. Medición de distancia con movimiento rotacional



Figura 9. Medición experimental de distancias con el dispositivo. Vista completa.



Figura 10. Marca de inicio.

8. Medición de distancia con movimiento rotacional

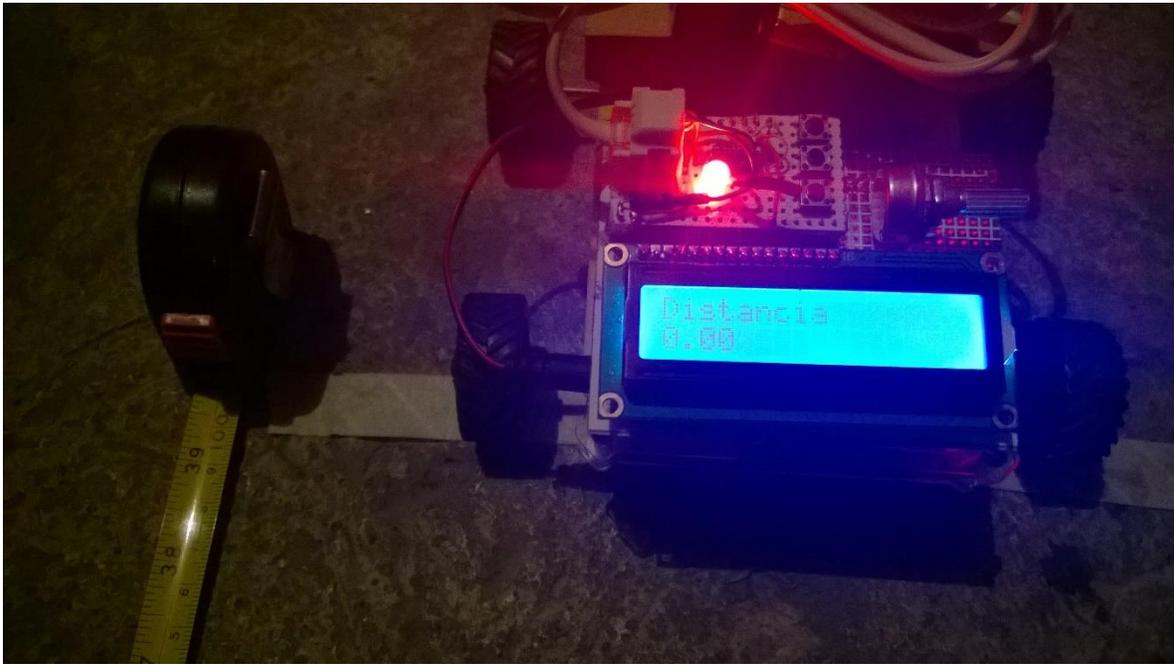


Figura 11. Inicialización de medición en el dispositivo.



Figura 12. Fin de la medición.

8. Medición de distancia con movimiento rotacional

RESULTADOS

Cálculo de la distancia con la rueda del carrito

El dispositivo funciona correctamente, pero con un error de 3 cm debido a que éste es el intervalo mínimo de detección que corresponde a la distancia recorrida por la rueda cuando el sensor detecta el siguiente agujero.

Hubo problemas para hacer las mediciones en lugares iluminados con el Sol debido a que el sensor infrarrojo se activa con esta luz y falla al hacer las lecturas.

El perímetro de la circunferencia es $C=D \times \pi$, así que en una vuelta completa el carrito recorre $(0.076 \text{ m}) \pi \approx 24 \text{ cm}$. Considerando que la rueda está subdividida en 8 partes iguales, la distancia entre ellas es $(0.076 \text{ m})\pi / 8 \approx 3 \text{ cm}$. Si la distancia a medir es 1 m, la rueda del carrito deberá dar $1/(0.076\pi)$ vueltas = 4.18 vueltas = 4 vueltas y casi los 2/8 de vuelta.

Se colocaron dos cintas separadas 1 metro y se utilizó el carrito para medir la distancia. El resultado de la pantalla fue un valor entre 0.99 m y 1.02 m porque sólo puede incrementar distancias cada 3 cm.

El experimento se llevó a cabo con alumnos de Física 1. El resultado fue muy positivo en el aspecto motivacional. Algunos de ellos mostraron interés en desarrollar su propio proyecto con Arduino.

En Física 3 no hubo oportunidad de probarlo con alumnos; sin embargo, ya se había hecho una práctica similar utilizando una bicicleta para medir.

DISCUSIÓN Y CONCLUSIONES

Motivación: Los alumnos mostraron gran interés en el dispositivo y les gustó mucho hacer las mediciones con el carrito.

Posible desarrollo por parte del alumno. Existen grandes posibilidades de que este experimento motive a los alumnos que cursan la materia de Física 3 a construir sus propios dispositivos de medición con Arduino, debido a que algunos de ellos tuvieron experiencias con este procesador en el taller de cómputo.

Movimiento rotacional y desplazamiento lineal. Este dispositivo es una excelente herramienta para comprender la relación entre movimiento rotacional y desplazamiento lineal.

Arduino vs. Tracker. En un principio se pensó utilizar la rueda con el brazo sujetador como polea (se puede hacer esto retirando la goma de hule de la rueda) para medir aceleración. Sin embargo, se encontró que es didácticamente más significativo que los alumnos graben un video y lo analicen con Tracker, razón por la cual no se continuó con el desarrollo de esta idea.

Aspectos por mejorar

Por el momento el dispositivo funciona en lugares que no cuentan con gran iluminación. Se requiere hacer pruebas de colocar una cubierta para aislarlo de las fuentes luminosas que interfieren con el sensor infrarrojo.

Posibles aplicaciones futuras

8. Medición de distancia con movimiento rotacional

Cuenta vueltas. Se podría aprovechar la rueda con los sensores para construir un cuenta vueltas útil para otras aplicaciones, como por ejemplo la construcción de transformadores.

Medición de rapidez rotacional. Para futuros desarrollos se podrían utilizar la rueda con el sensor para medir rapidez rotacional de objetos físicos como un taladro, una rueda de bicicleta, un generador eólico, etc.

Medición de alta velocidad, aceleración y dirección. Se habilitó la conexión del motor, que por el momento solo funciona como eje de rotación, pensando que podría aprovecharse como sensor para hacer mediciones de movimiento rotacional considerando la proporcionalidad entre rapidez de rotación y voltaje cuando el motor se utiliza como un generador. Este también sería capaz de detectar dirección de movimiento, ya que la polaridad del voltaje generado se invierte si se cambia el sentido de la rotación.

REFERENCIAS

Bañuelos, M. A. (2017). Tutorial sobre Arduino. México: ICAT-UNAM.

Giancoli (2009). Física, Principios con aplicaciones. México: Pearson / Prentice Hall

8. Medición de distancia con movimiento rotacional

ANEXOS

1.- Programa que se debe grabar en Arduino para hacer el experimento

```
/*
*****
*/
/* Programa para medir distancias en metros con una rueda de 8 agujeros */
/* un Opto Sensor QRD1114 y Display de cristal líquido */
*****
*/
```

Notas:

La terminal positiva del optosensor se conecta a la entrada analógica A0. De esta manera se logró solucionar el problema de las oscilaciones en la transición al definir un intervalo acotado por LimSup y Lim Inf.

El sensor QRD1114 es sensible a la luz ordinaria, por lo que deberá colocarse una pantalla para impedir que esta llegue al sensor. El sensor no funciona bajo la luz directa del Sol

```
#include <LiquidCrystal.h>
```

```
/** Definiciones **/
int const pinOpto = A0; //Optosensor
int const pinLed = 8; //Led indicador
int const PinReset = 9; //Botón para inicializar
int const LimSup = 700;
int const LimInf = 300;
int i=0;
int EstadoBotonReset=0;
float LecturaActual;
float LecturaAnterior;
float Distancia=0;
```

```
// Inicialización de la biblioteca con la definición de los pines de conexión
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
```

```
void setup() {
```

```
    // Inicialización de los pines
    pinMode(pinLed, OUTPUT);
    pinMode(PinReset, INPUT);
    Serial.begin(9600);
```

```
    // Lee el opto y asigna estado
    LecturaActual = analogRead(pinOpto);
```

8. Medición de distancia con movimiento rotacional

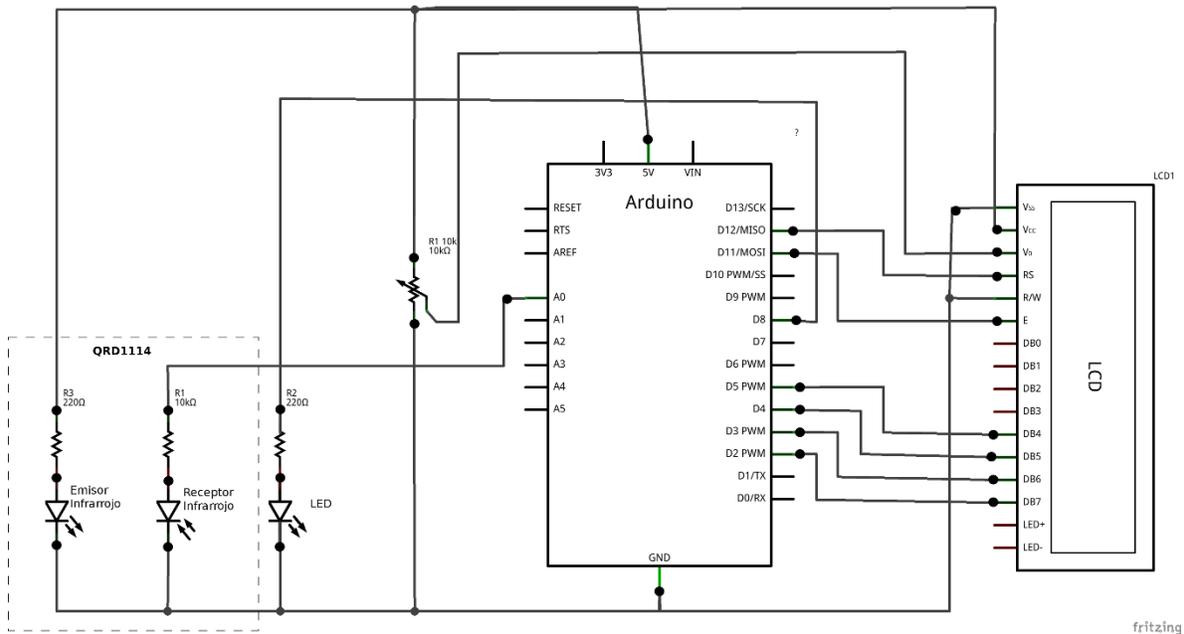
```
if (LecturaActual > LimSup) digitalWrite(pinLed, HIGH);  
else digitalWrite(pinLed, LOW);  
LecturaAnterior=LecturaActual;
```

```
// Se establece el número de columnas y renglones  
lcd.begin(16, 2);  
// Muestra un mensaje en la pantalla LCD  
lcd.print("Distancia");  
lcd.setCursor(0, 1);  
lcd.print(Distancia);  
}
```

```
void loop() {  
  // Programa principal  
  LecturaActual = analogRead(pinOpto); // Lee el optosensor  
  EstadoBotonReset = digitalRead(PinReset);  
  
  // Identifica si cambió la lectura del optosensor  
  if (LecturaAnterior>LimSup && LecturaActual<LimInf || LecturaAnterior<LimInf &&  
LecturaActual>LimSup)  
  {  
    LecturaAnterior=LecturaActual;  
    i++;  
    if (LecturaActual>LimSup)  
    {  
      digitalWrite(pinLed, HIGH);  
      Distancia += 3.141516*0.076076/8;  
    }  
    else digitalWrite(pinLed, LOW);  
  
    // Selecciona la columna 0, renglón 1  
    // (Nota: la línea 1 es el segundo renglón, ya que el primero es la línea 0  
    lcd.setCursor(0, 1);  
    lcd.print(Distancia);  
  }  
}
```

8. Medición de distancia con movimiento rotacional

2.- Diagrama esquemático del circuito electrónico que se conecta al Arduino



3.- Observaciones sobre el algoritmo del programa utilizado

Se presentaron severas dificultades para la lectura del sensor infrarrojo en la entrada digital de Arduino debido a que en la transición entre los estados recibir / no recibir señal manda una señal de pulsos que Arduino interpreta como la ocurrencia de varios eventos en lugar de uno solo. Este error se solucionó utilizando la entrada analógica y definiendo valores máximos y mínimos para decidir la ocurrencia o no ocurrencia del evento, debido a que existen valores máximos y mínimos del voltaje que envía el sensor cuando recibe o no recibe señal.

Se tuvieron que minimizar las operaciones del procesador Arduino, ya que de lo contrario se satura y deja de percibir eventos cuando la rueda gira muy rápido.

9. DETERMINACIÓN DE ÍNDICE ULTRAVIOLETA CON ARDUINO

Responsable: Sergio Alejandro Carrillo Araujo. CCH-Sur	Participante: Eduardo José Vega Murguía ICAT	Revisión: Eduardo José Vega Murguía ICAT
--	--	--

Ubicación Curricular:

Colegio de Ciencias y Humanidades

Física II, Área de Ciencias Experimentales

Unidad 2. Ondas: mecánicas y electromagnéticas

Resumen: Se presenta una estrategia para determinar el índice de radiación ultravioleta (IUV) aplicando el microcontrolador Arduino UNO y el sensor UVM 30A. Se determina el índice de radiación ultravioleta mediante una serie de mediciones directas con lámparas UV y con el UV solar. Comparamos nuestros resultados con los de las bases de datos de las estaciones meteorológicas del plantel del CCH Sur. Se aplicó el uso de microcontroladores y sensores en el bachillerato con la finalidad de ampliar el horizonte cultural de los estudiantes. Los conceptos de la unidad 2 del programa de Física II sobre el espectro electromagnético pueden ser abordados mediante una práctica que refiere la importancia de la radiación UV.

Palabras clave: Espectro electromagnético, radiación ultravioleta, sensores, Arduino.

INTRODUCCIÓN

*GUÍA PARA EL PROFESOR: En el programa de Física del Colegio de Ciencias y Humanidades se aborda el tema del **espectro electromagnético** en la segunda unidad del programa de estudios de Física II (CCH UNAM, 2003). Se encuentran incorporados los temas de Ondas electromagnéticas y su espectro. Los aprendizajes estipulados en el programa indican que el alumno debe describir el espectro de ondas electromagnéticas e identificar a la luz visible como parte de él. Para comprender la luz como parte de él se debe explorar el acotamiento superior e inferior de la frecuencia, lo que implica el estudio por encima del violeta y por debajo del rojo (zonas del ultravioleta e infrarrojo) (Bueche, 1999), (Resnick, Halliday y Krane, 2001).*

A mediados del siglo XIX la teoría electromagnética de James Clerk Maxwell permitió la cabal comprensión de la luz como fenómeno ondulatorio. Se reconoció que la luz blanca comprende al conjunto de colores que observamos en un arco iris (espectro visible) y se expandió la comprensión de las ondas electromagnéticas hasta lo que hoy llamamos espectro electromagnético. Las contribuciones por un lado con William Herschel, quien fue el primer astrónomo en detectar que había cierta cantidad de calor por debajo de la parte roja del espectro, descubriendo así el infrarrojo por el año de 1800. Por otra parte J. Wilhelm Ritter en 1801 descubrió que por la parte más allá del violeta del espectro también existen unos rayos invisibles capaces de oscurecer granos de sales de plata puestos en papel. Se llamó a estos últimos “rayos desoxidantes” (ultravioleta) para distinguirlos de los

9. Determinación de índice ultravioleta con Arduino

rayos calóricos de Herschel. Comienza de esta manera a completarse el cuadro del espectro electromagnético.

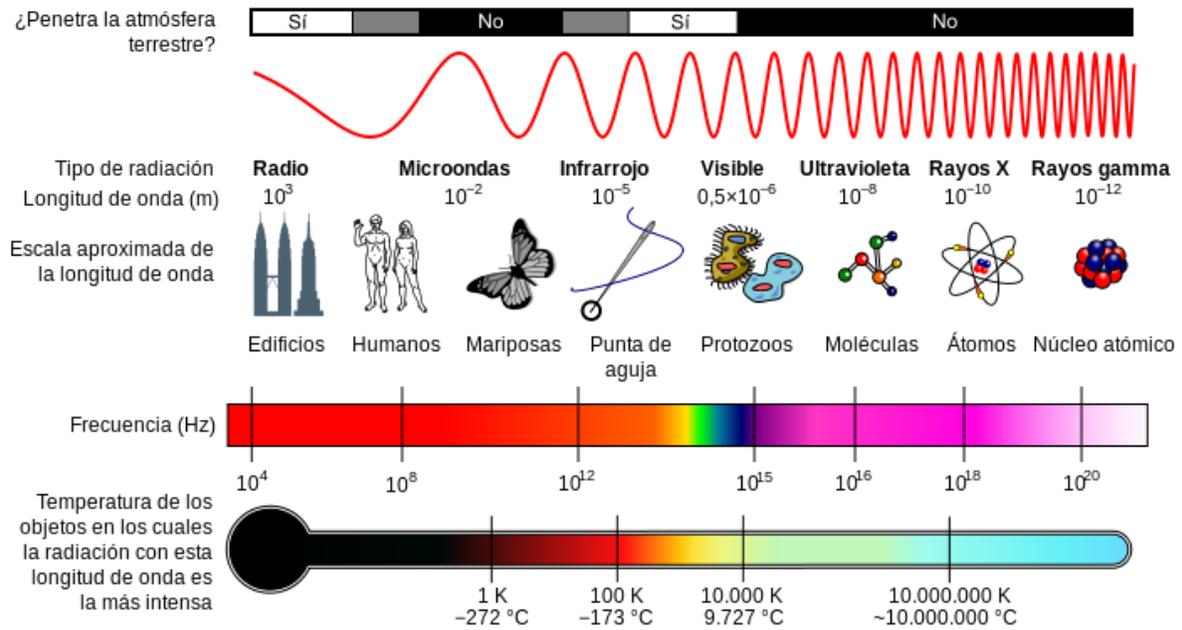


Figura 1. Espectro electromagnético incluye desde las ondas de radio hasta los rayos gamma.

Fuente: Wikipedia.

La radiación ultravioleta se piensa regularmente como una radiación dañina; sin embargo, es pertinente aclarar que resulta necesaria la exposición a ciertas dosis de esta radiación para poder sintetizar la vitamina D. También resulta que no todas las personas presentan la misma tolerancia a la exposición de radiación solar UV. Las personas con tonos de piel más oscuros son más tolerantes a la exposición (UV), mientras que las personas con piel clara son más sensibles. Los estudios llevados a cabo por médicos especialistas en enfermedades de la piel y ojos permitieron establecer un indicador de radiación solar UV recomendado por la OMS, abreviada por las siglas IUUV, donde se estiman los valores aceptables de exposición a la radiación UV (OMS, 2014). Se establece como valores aceptables un índice de hasta 7 donde cada unidad del índice corresponde a una radiación de $0.025 \text{ W/m}^2 = 1\text{MED}$.

9. Determinación de índice ultravioleta con Arduino



Figura 2. Índice UV tomado de la página de la CDMX [5]. Tomada de: <http://www.aire.cdmx.gob.mx/default.php?opc=%27ZaBhnml=&dc=%27aA>.

En esta práctica se estudia el espectro electromagnético asociado a la radiación solar UV. Se programa y diseña una placa de Arduino UNO con un sensor UV (UVM 30A). Se determina la radiación solar en un día de preferencia soleado y se compara el resultado con lo que obtienen los sensores de UV del programa de estaciones meteorológicas del bachillerato (UNAM, 2019). De esta manera se sensibiliza a los estudiantes de los efectos a la exposición de radiación ultravioleta.

OBJETIVOS

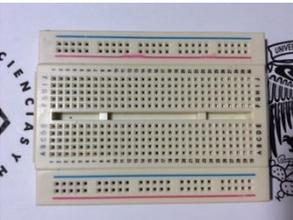
A través de implementar un dispositivo con Arduino que pueda medir el índice de radiación ultravioleta solar, los alumnos describirán el espectro de ondas electromagnéticas e identificarán a la luz visible como parte de él. También deberán describir algunos usos y aplicaciones de las ondas electromagnéticas.

El objetivo particular es comparar la emisión de radiación UV de diferentes lámparas y la proveniente del Sol.

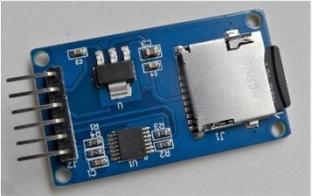
MATERIAL

Descripción	Imagen	Comentarios
Microcontrolador Arduino UNO		Hay diferentes tipos de Arduino y el más común es el UNO.

9. Determinación de índice ultravioleta con Arduino

Descripción	Imagen	Comentarios
Cable de alimentación USB		Regularmente USB A macho a USB B macho.
Pantalla de Cristal Líquido LCD 16x32 (1602A)		Con led de fondo azul o verde. Hay pantallas sin Led de fondo son más baratas y funcionan casi del mismo modo. Se requiere que el usuario suelde los pines de conexión.
Potenciómetro de 10 kΩ		Resistencia variable para controlar la intensidad de la pantalla LCD. Los extremos van conectados a tierra y la alimentación.
Protoboard de 400 puntos		Existen de diferentes tamaños. Uno pequeño está bien para estos proyectos.
Jumpers macho-macho		Se necesitan varios, se recomienda adquirir por lote de 40 cables.
Jumpers macho-hembra		Se necesitan varios, se recomienda adquirir por lote de 40 cables.
Sensor de ultravioleta UVM30A		Sensor que mide la intensidad de radiación UV. Existe una regla de asociación del voltaje asociado con cada índice UV.

9. Determinación de índice ultravioleta con Arduino

Descripción	Imagen	Comentarios
Módulo lector de memoria microSD		Se recomienda para usar con una memoria de almacenamiento microSD.
Computadora con programa Arduino		Se recomienda tener instalada la última versión estable de Arduino.
Batería de respaldo de 2000 mAh		Existen diversas marcas. Revisar que estén cargadas.

Software

- Programa Arduino IDE 1.8.2 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>.
- Sketch, proporcionado por el profesor.
- Hoja de cálculo: Excel, Graph, Numbers o Datasheet.
- Programa Fritzing 0.9.3b o superior. Disponible en <http://fritzing.org/download/>.

GUÍA PARA EL PROFESOR:

Indispensable el software Arduino IDE para la programación de instrucciones para el microcontrolador desde una computadora personal.

El manejo de datos se puede realizar en hojas de datos o programas para graficar (Excel, Graph, Numbers o Datasheet).

Se pueden hacer esquemas de ensamblaje con el programa Fritzing que es software abierto. Es recomendable para que los alumnos manejen las conexiones de los sensores de forma virtual.

El sensor UVM30A no requiere bibliotecas especiales, pero se puede consultar las referencias (EKT) y (SGLUX).

Algunos applets para ilustrar con simulaciones el concepto de radiación electromagnética y espectro electromagnético, en particular se recomienda visitar los de la universidad de Colorado PhET (Colorado).

9. Determinación de índice ultravioleta con Arduino

DESARROLLO

1.- En esta práctica se debe tomar en cuenta previamente las siguientes consideraciones:

- Armar y programar las placas de Arduino con los sensores indicados (UVM30A).
- Cargar el programa que será manejado por la placa (el microcontrolador, Arduino UNO).
- Asegurarse que las conexiones sean correctas sobre todo en los sensores, pues se corre el riesgo de quemaduras, cortos circuitos y daño al equipo.
- Contar con un equipo de cómputo funcional para programar las instrucciones al microcontrolador.
- Revisar la hoja de especificaciones del sensor UVM30A.

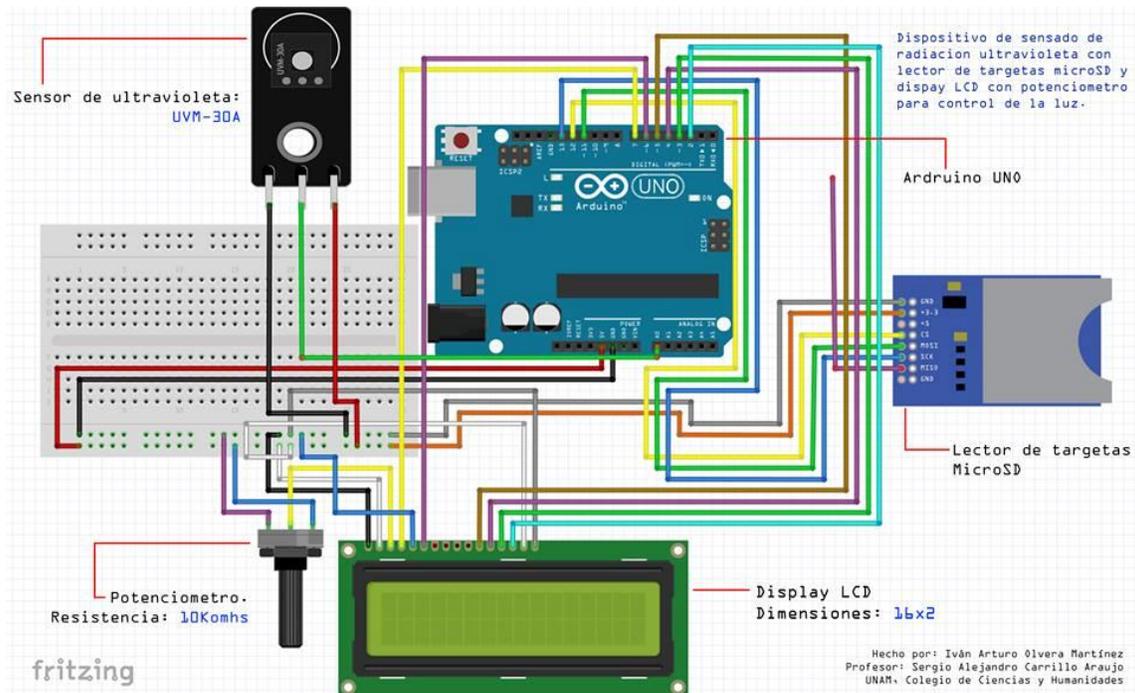


Figura 3. Esquema de conexión del microcontrolador, pantalla LCD, tarjeta de datos Micro SD y el sensor de UV. Observe que el sensor UV se conecta al pin A0 del Arduino. El lector de datos permite tomar mediciones y guardarlas en una memoria microSD.

Actividades

2.- **Actividad 1.** Caracterizar el instrumento de medición.

Investigue y elabore un resumen de cómo funcionan el sensor empleado para las mediciones SGLUX () y UNAM (2014). Llene la siguiente tabla.

Instrumento	Rango de operación	Resolución	Observación
UVM30A			
Otro sensor (opcional)			

9. Determinación de índice ultravioleta con Arduino

Mida el ultravioleta emitido por lámparas caracterizadas como las que hay en el laboratorio y/o el ultravioleta emitido por los focos ahorradores, focos de filamento, leds UV. Compare el valor de sus mediciones.

Puede hacer un comparativo de la intensidad de radiación recibida en función de la distancia a las lámparas. ¿Qué tipo de relación espera?

Para registrar su información puede hacer una tabla donde indique para cada lámpara cuánta radiación UV emite a diversas distancias del sensor a la lámpara.

3.- **Actividad 2.** Registre sus datos y ajuste un modelo para describir el comportamiento del IUV solar. Para esta actividad será necesario ahorrar energía apagando el LED de fondo de la pantalla LCD. Use baterías que le permitirán capturar información entre 6 y 8 hrs. Deje transcurrir un tiempo considerable entre cada lectura de IUV se recomienda tomar lecturas cada 15 o 30 minutos. La lectura de datos se puede observar en la pantalla LCD del dispositivo, pero como estarán almacenadas en memoria microSD se requiere de una computadora para transferirle los resultados y graficar los resultados. Si se manda los datos vía bluetooth o WIFI se podrá realizar una gráfica en tiempo real.

El índice UV presenta una variabilidad debida a la rotación de la Tierra, pero también depende de las condiciones climáticas del cielo. Regularmente se ve una distribución con picos máximos entre las 12 y las 3 de la tarde con mínimos al amanecer y al atardecer.

RESULTADOS

Las especificaciones del sensor UVM30A señalan una precisión de ± 1 IUV. Y la siguiente información:

- Working voltage: DC 3~5V
- Current: 0.06mA(Standard)/0.1mA(Max)
- Response wavelength: 200~370nm
- Working temperature: -20~85°C

El sensor tiene una respuesta aproximadamente lineal, como se muestra en la figura 4.

9. Determinación de índice ultravioleta con Arduino

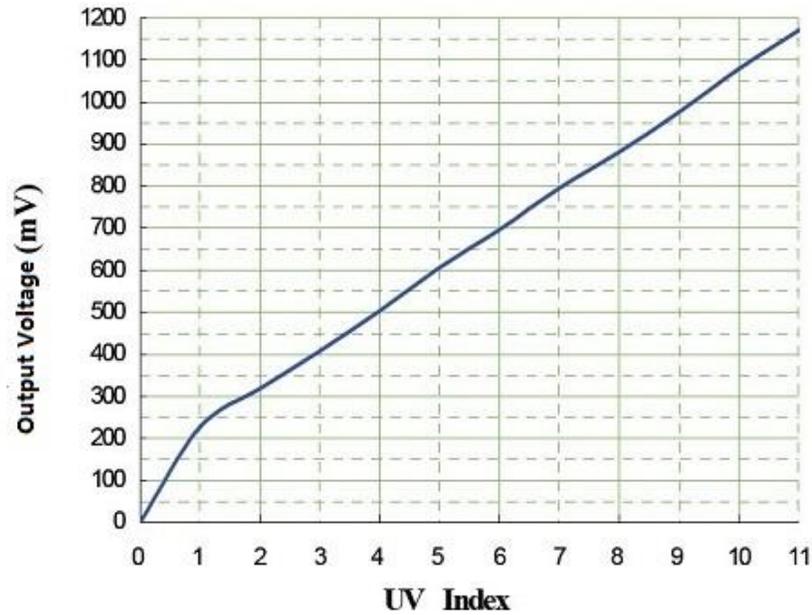


Figura 4. Respuesta en voltaje del sensor UVM30A en función del índice de radiación ultravioleta. Tomada de: <https://sites.google.com/site/myterrarium23/domotique/uvm-30a-uva-uvb-sensor>.

La tabla de valores del IUV, que se muestra en la figura 5, asocia los siguientes valores de acuerdo con el voltaje registrado.

紫外线指数 UV Index	0					
Vout(mV)	<50	227	318	408	503	606
紫外线指数 UV Index						
Vout(mV)	696	795	881	976	1079	1170+

Figura 5. Tabla del Índice de radiación Ultravioleta. Tomada de: <https://sites.google.com/site/myterrarium23/domotique/uvm-30a-uva-uvb-sensor>.

En la página de los parámetros meteorológicos de la CDMX (SEDEMA), se pueden encontrar los reportes de última hora, así como información sobre las mediciones del IUV. Por ejemplo, en la figura 6 se presenta el reporte del 28 de marzo de 2017.

9. Determinación de índice ultravioleta con Arduino

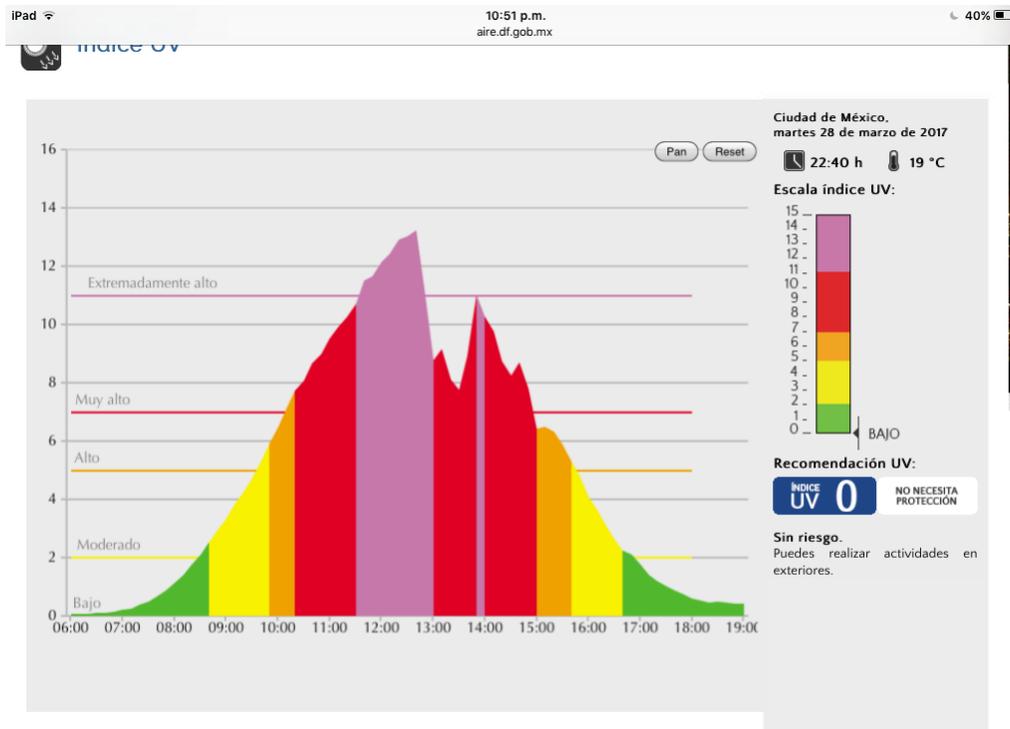


Figura 6. Reporte del índice de radiación ultravioleta tomado de la página de la CDMX.

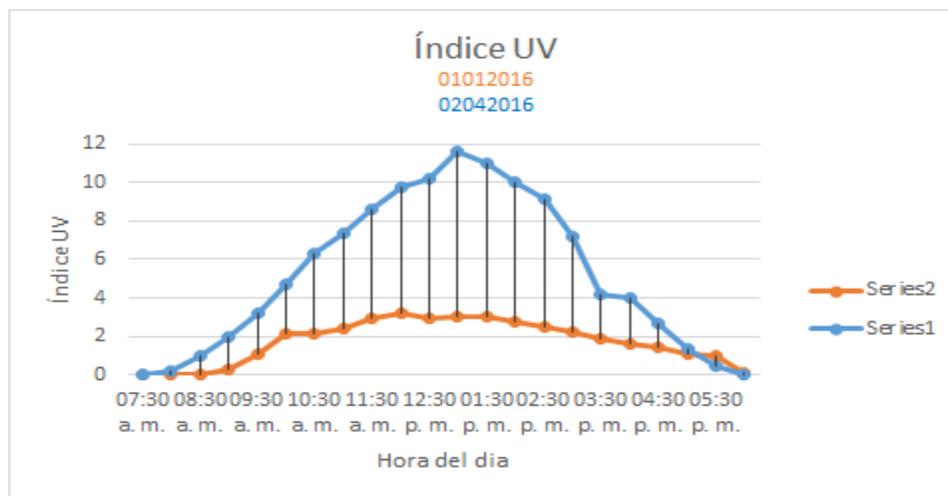


Figura 7. Mediciones del IUV obtenidas por PEMBU. Se espera obtener gráficos muy similares a las distribuciones mostradas.

DISCUSIÓN Y CONCLUSIONES

Conteste las siguientes preguntas y justifique sus respuestas.

- ¿Qué es lo que mide el sensor UVM30A?
- ¿Cómo se comporta la relación entre la intensidad de radiación UV y la distancia a la fuente?
- ¿Qué representa el índice de radiación UV?
- ¿Cómo es el comportamiento de la intensidad de radiación UV solar medida a lo largo del día?

9. Determinación de índice ultravioleta con Arduino

- ¿De qué depende la intensidad de radiación UV solar medida aquí en la tierra?
- ¿Cómo es la tolerancia a la exposición de la radiación UV para las personas?
- ¿Es siempre dañina la radiación UV?
- Redacte unas breves conclusiones sobre los beneficios y daños producidos por a la radiación UV.

GUIA PARA EL PROFESOR: Por el momento se tienen resultados preliminares de la determinación del IUV, nuestro sensor es capaz de medir y estimar el parámetro de IUV para comparar con los valores registrados por algunas estaciones meteorológicas de la CDMX y PEMBU.

Los estudiantes pueden intentar sacar variaciones del IUV por mes o incluso por año pues hay muchas fuentes que proporcionan los datos. Pongo los datos de PEMBU como ejemplo la que se muestra en la figura 8.

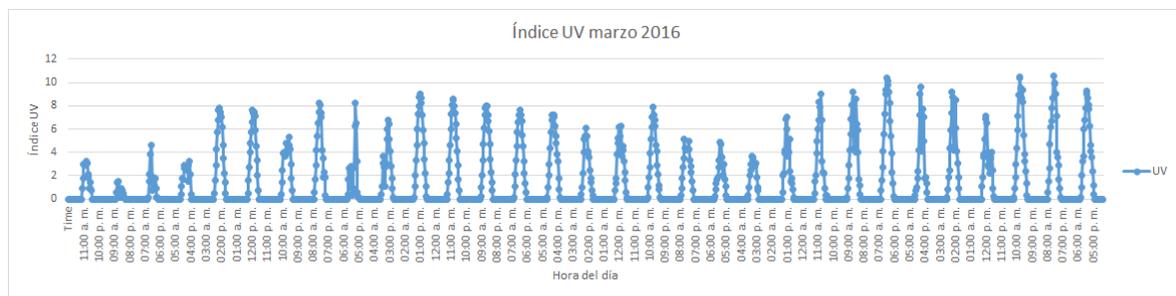


Figura 8. Los valores del IUV a lo largo de un mes. Se pueden observar algunas variaciones que pudieran tener correlación con la actividad directa del Sol. Parece ser que en gran medida los procesos de liberación de energía en el sol junto con su actividad superficial pudieran estar influenciando directamente el comportamiento del clima en la tierra.

Los estudiantes cuentan con un diseño de un sensor que les permitirá llevar a cabo prácticas que involucran a la radiación ultravioleta. Una posible sería determinar la efectividad de los filtros solares para la radiación UV.

El uso de sensores puede fomentar la creatividad de los jóvenes, así como la capacidad de innovar en ciencias experimentales. Algunas prácticas de laboratorio pueden apoyarse en los sensores y microcontroladores para obtener mediciones de parámetros físicos.

Con este tipo de prácticas se advierte a los estudiantes para que desarrollen conciencia de la exposición que tenemos a la radiación UV. Se comprenden sus beneficios como sus factores de riesgo por exposición prolongada.

REFERENCIAS

Bueche, F. J. (1991). Fundamentos de física. Tomo II. México: McGraw Hill.

CCH UNAM (2003) Plan de estudios física I a IV. Recuperado el 6 de junio de 2019 de http://www.cch.unam.mx/sites/default/files/plan_estudio/mapa_fisica.pdf

Colorado, U. (). Blackbody Spectrum. Recuperado el 6 de junio de 2019 de

9. Determinación de índice ultravioleta con Arduino

<https://phet.colorado.edu/en/simulation/legacy/blackbody-spectrum>

EKT (). UVM -30A datasheet . Recuperado el 6 de junio de 2019 de http://www.ekt2.com/pdf/412_ARDUINO_SENSOR_ULTRA_VIOLET.pdf

Gallardo, S. (2014). UVM-30A UVA/UVB sensor. Recuperado el 6 de junio de 2019 de <https://sites.google.com/site/myterrarium23/domotique/uvm-30a-uva-uvb-sensor>

Nave, R (). The Electromagnetic Spectrum. Recuperado el 6 de junio de 2019 de <http://hyperphysics.phy-astr.gsu.edu/hbase/hframe.html>

OMS (2003). Índice UV solar mundial. Organización mundial de la salud. Recuperado el 6 de junio de 2019 de <http://www.who.int/uv/publications/en/uvispa.pdf>

Resnick, R., Halliday, D. y Krane, K. (2001). Física Vol. 2. México: CECSA.

SEDEMA (), Índice de radiación UV CDMX. Recuperado el 6 de junio de 2019 de <http://www.aire.cdmx.gob.mx/default.php?opc=%27ZaBhnml=&dc=%27aA>

SGLUX (). GUVA-S10GD Ultraviolet sensor. Recuperado el 6 de junio de 2019 de <http://www.electronicaestudio.com/docs/SHT-163hd.pdf>

UNAM (2014). Enciclopedia conocimientos fundamentales UNAM Siglo XXI. Recuperado el 5 de junio de 2019 de <http://www.unamenlinea.unam.mx/recurso/enciclopedia-de-conocimientos-fundamentales-unam-siglo-xxi-matematicas>

_____ (2019). Programa de estaciones meteorológicas del bachillerato universitario PEMBU. Recuperado el 6 de junio de 2019 de <http://www.ruoa.unam.mx/pembu/CCHS.html>

9. Determinación de índice ultravioleta con Arduino

ANEXOS

```
/******  
* Autor: Sergio Alejandro Carrillo Araujo *  
  Ivan Olvera *  
* CCH Sur UNAM *  
* CC:3.0 *  
* 16/11/2016 Última revisión: 21/03/2017 *  
* Proyecto: Sensor de UV *  
* Estado: Funcional *  
* Material: Arduino Leonardo, UNO, Display 1602A, *  
* Sensor UVM 30A ,Potenciómetro 10K *  
* Proposito: Medir índice UV *  
* *  
* Detalles: Hay que calibrar el sensor *  
*****/  
//Bibliotecas  
#include <LiquidCrystal.h>  
#include <SD.h>  
#include <SPI.h>  
  
#define PIN_GUVAS12SD A0 // Pin al que se conecta el módulo con el UVM30A  
#define ESPERA_ENTRE_LECTURAS 100 // Leer cada 100 ms  
#define ESPERA_ENTRE_PRESENTACIONES 30000 // Mostrar el índice cada 30 s  
#define CANTIDAD_INDICES_UV 11  
  
File myfile;  
  
unsigned int lectura_sensor;  
unsigned int contador_lecturas=0;  
float media_lecturas=0.0;  
int valor_indice_uv[CANTIDAD_INDICES_UV]={210,295,378,467,563,646,738,818,907,1003,1022}; // De 1 a 11  
byte indice;  
boolean buscando_indice_uv;  
long cronometro_lecturas;  
long cronometro_presentaciones;  
long tiempo_transcurrido;  
  
LiquidCrystal lcd(7,6,5,4,3,2);  
/#####  
void setup()  
{  
  analogReference(INTERNAL); // Referencia interna de 1100 mV El GUVAS-12SD mide de 0 a 1170 mV que corresponde con índice UV 11  
  //pinMode(PIN_GUVAS12SD,INPUT); // La lectura analógica no necesita inicialización  
  Serial.begin(9600);  
  
  lcd.begin(16,2);  
  lcd.setCursor(0,0);  
  // Escribir primer mensaje  
  lcd.print(" Buen dia Alex");  
  lcd.setCursor(0,1);  
  lcd.print(" UNAM CCH SUR");  
  
  #if defined (__AVR_ATmega32U4__)  
    while(!Serial); // Esperar a Arduino Leonardo  
  #endif  
  lectura_sensor=analogRead(PIN_GUVAS12SD); // La primera lectura es incorrecta (normalmente cero)  
  cronometro_lecturas=millis(); // Esperar un ciclo de lectura para estabilizar el sensor y la entrada analógica  
  cronometro_presentaciones=millis();  
  Serial.println("Hola Alex");  
  Serial.println("Iniciando Tarjeta SD");  
  if(!SD.begin(4)){  
    Serial.println("No se pudo iniciar Tarjeta SD");  
    return;  
  }  
}
```

9. Determinación de índice ultravioleta con Arduino

```
}

/#####
void loop()
{
  lcd.setCursor(16,1);
  lcd.autoscroll();
  lcd.print(" ");
  delay(300);

  tiempo_transcurrido=millis()-cronometro_lecturas;
  if(tiempo_transcurrido>ESPERA_ENTRE_LLECTURAS)
  {
    cronometro_lecturas=millis();
    lectura_sensor=analogRead(PIN_GUVAS12SD);
    contador_lecturas++;
    media_lecturas=(float)lectura_sensor/(float)contador_lecturas+media_lecturas*(float)(contador_lecturas-1)/(float)(contador_lecturas);
  }
  tiempo_transcurrido=millis()-cronometro_presentaciones;
  if(tiempo_transcurrido>ESPERA_ENTRE_PRESENTACIONES)
  {
    cronometro_presentaciones=millis();
    buscando_indice_uv=true;
    indice=CANTIDAD_INDICES_UV;
    while(buscando_indice_uv&&indice>0)
    {
      indice--;
      if(media_lecturas>valor_indice_uv[indice])
      {
        buscando_indice_uv=false;
      }
    }
    Serial.print(contador_lecturas);
    Serial.print("Lectura sensor: "+String(media_lecturas,DEC));
    Serial.print(" (media de "+String(contador_lecturas,DEC)+"");
    Serial.println(" Indice UV: "+String(indice,DEC));

    lcd.setCursor(0,1);
    lcd.print(" Indice UV: "+String(indice,DEC));
    // lcd.scrollDisplayRight();
    // delay(150);
    contador_lecturas=0;
    media_lecturas=0.0;
  }
  myfile=SD.open("datalog.txt",FILE_WRITE);
  if(myfile){
    Serial.print("Escribiendo en SD ");
    int uvLevel=analogRead(0);
    myfile.print("Tiempo(ms)= ");
    myfile.print(millis());
    myfile.print("UVLevel= ");
    myfile.println(uvLevel);
    myfile.close();
    Serial.print("UVLevel= ");
    Serial.println(uvLevel);
    delay(1500);}else{
    Serial.println("Error al abrir el archivo");
  }
}
/##### FIN #####
```

10. MIDIENDO LA TEMPERATURA EMPLEANDO EL ARDUINO PARA AYUDAR A IDENTIFICAR PAREDES AISLANTES Y DIATÉRMICAS

Responsable: Ingrid Escobedo Estrada Facultad de Química	Participantes: Miguel Á. Bañuelos Saucedo ICAT	Revisión: Eduardo José Vega Murguía ICAT
--	--	--

Ubicación curricular

Colegio de Ciencias y Humanidades

Física I

Unidad 3. Energía: fenómenos térmicos, tecnología y sociedad

Colegio de Bachilleres

Física II

Contenido central: Conservación de la Energía

INTRODUCCIÓN

Si un científico se para enfrente de un adolescente en un día en el que la temperatura del medio ambiente es muy baja y le pregunta ¿la chamarra que traes puesta es una excelente pared adiabática?, lo más probable es que el joven o la joven lo vea con cara de interrogación, no le conteste, se dé la media vuelta y se vaya, ¿verdad? Pero la situación cambia si la pregunta ahora es, ¿la chamarra que traes es “calientita”?, o ¿tu chamarra “esta tan gruesa” que te protege bien del frío? En este caso, seguramente, y por lo menos, recibirá un sí o un no. Situaciones como ésta y similares, están presentes todo el tiempo en la vida de cualquier persona que habite este planeta. ¿Y qué con ello? Pues varias observaciones, por mencionar algunas: en primer lugar, muchas personas tienden a pensar que entre más gruesa es una chamarra, más “calientita” es; en segundo lugar, pocas veces se detiene uno a pensar en que el tipo de material del que esté hecho la chamarra también influye en que tanto proteja del medio ambiente o no; por último, siempre se piensa que la chamarra nos cubre del frío, y no que es nuestro cuerpo el que se aísla para que no ceda su calor al medio ambiente.

Es importante que los estudiantes de nivel medio superior se percaten de que existen por lo menos dos tipos de paredes: adiabáticas y diatérmicas, para que se utilicen dependiendo de las necesidades que se tengan.

OBJETIVO

El estudiante, con ayuda del Arduino observará la temperatura de varios trozos de hielo envueltos en una superficie diferente cada uno de ellos, para comprobar que no se derriten al mismo tiempo, y

10. Midiendo la temperatura empleando el Arduino para ayudar a identificar paredes aislantes y diatérmicas

que esto se debe a que la temperatura de cada sistema, tarda más o menos tiempo en cambiar, en función de las condiciones que le rodean.

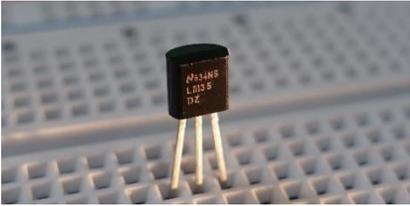
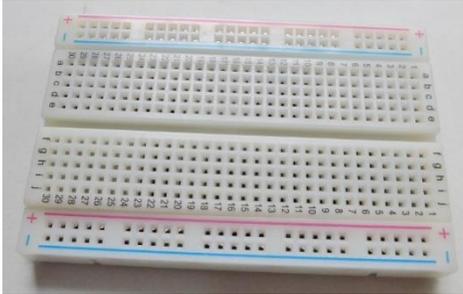
MATERIAL

Descripción	Imagen	Comentarios
8 cubos de hielo		Procurar que su tamaño sea lo más similar posible.
Una bufanda tejida		Si no se tiene a la mano una bufanda tejida, puede usarse una hecha de cualquier otro material. También funciona una toalla.
Una bolsa de plástico que sea capaz de contener sin problemas, a uno de los cubos de hielo.		
Un pedazo de papel periódico. Este debe de rodear completamente a uno de los cubos de hielo.		

10. Midiendo la temperatura empleando el Arduino para ayudar a identificar paredes aislantes y diatérmicas

Descripción	Imagen	Comentarios
Un pedazo de papel aluminio. Que también rodee a uno de los cubos de hielo		
Una cajita de madera. Para poner dentro de ella a otro de los cubos de hielo.		
Un vaso de Unicel con todo y su tapa también de unicel, como los del atole		
Una caja de zapatos para bebé o cualquier caja pequeña de cartón que contenga sin problemas al cubo de hielo		
Tierra de maceta, la suficiente para rodear por si misma a un cubo de hielo		

10. Midiendo la temperatura empleando el Arduino para ayudar a identificar paredes aislantes y diatérmicas

Descripción	Imagen	Comentarios
Un sensor de temperatura LM35 o DS18B20		Cualquiera de los dos sensores es útil para el desarrollo de la actividad
Una tarjeta ARDUINO UNO		
Cable USB A macho a B macho		
Una resistencia de 1 kΩ (café, negro, rojo)		Es opcional dependiendo del funcionamiento del circuito
Un protoboard		

10. Midiendo la temperatura empleando el Arduino para ayudar a identificar paredes aislantes y diatérmicas

Descripción	Imagen	Comentarios
Trimpot de 10 kΩ		Se puede sustituir por un potenciómetro miniatura (como el ilustrado).
O en lugar de la laptop, si el programa ya se cargó en el Arduino, se puede recurrir a una pila de 9 V con todo y su conector		Conector de broche para pila de 9 V con adaptador plug invertido.
Un display LCD de 16x2 caracteres		El usuario debe soldar los pines de conexión.

GUÍA PARA EL PROFESOR: El profesor puede elegir con qué tipo de materiales desea rodear a cada cubo. Los mencionados solo son para dar un ejemplo. Lo único que se sugiere es que por lo menos la mitad de la cantidad de materiales a utilizar sean aislantes (adiabáticos) y la otra mitad diatérmicos.

Software

- Programa Arduino IDE 1.8.2 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>.
- Programa CoolTerm 1.4.7 o superior. Disponible en: <http://freeware.the-meiers.org/>.
- Programa LiquidCrystal.h. Disponible en: <https://www.arduino.cc/en/Main/Software>.

GUÍA PARA EL PROFESOR: el programa de LiquidCrystal.h en realidad es una biblioteca contenida dentro del programa que hace funcionar el Arduino, pero para que la pantalla LCD muestre la lectura registrada por el microprocesador hay que cargarla en el programa realizado. Es importante, que se cargue el programa de manera anticipada en todos los Arduinos con los que se vaya a realizar la actividad experimental, de lo contrario se puede perder tiempo valioso. Si el sensor utilizado es el LM35, hay que ponerle una extensión a las patitas del mismo, como lo muestra la figura 1, también un poco de cinta adhesiva, cinta de aislar o pegamento rápido pueden servir.

10. Midiendo la temperatura empleando el Arduino para ayudar a identificar paredes aislantes y diatérmicas

El Sketch con el que se programó al Arduino se presenta en los anexos, al igual que el diagrama de conexiones.

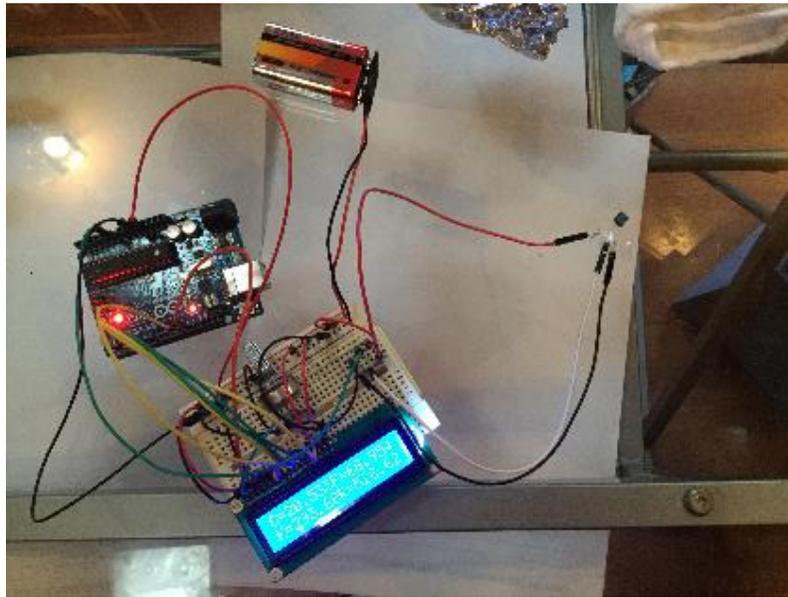


Figura 1. Fotografía del instrumento implementado.

DESARROLLO

1. Se coloca o se rodea un cubo de hielo dentro y por cada una de las superficies aportadas por los estudiantes, la bufanda, la bolsa de plástico, etc. Como lo muestra la figura 2.

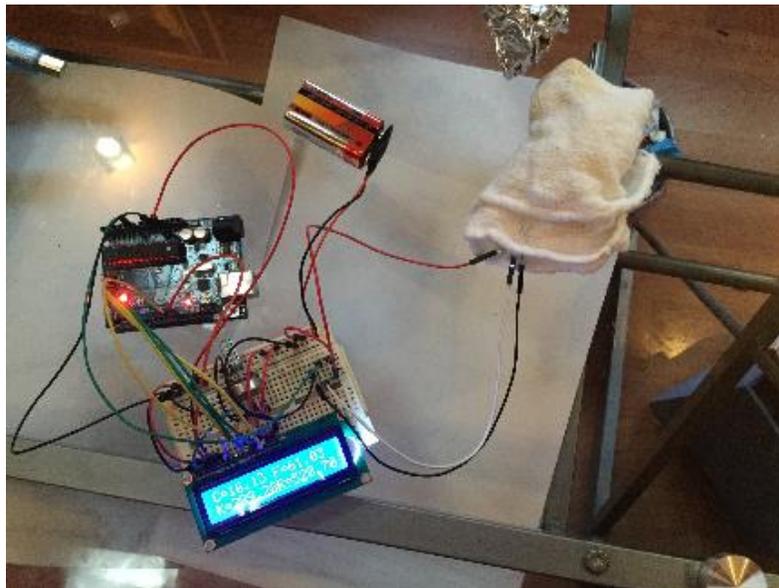


Figura 2. Medición de la temperatura de un cubo de hielo envuelto por un material textil.

2. Cada sistema se deja expuesto al medio ambiente. A los tres minutos se observa el estado de cada cubo de hielo y se mide su temperatura con ayuda del sensor y del Arduino. Sin embargo,

10. Midiendo la temperatura empleando el Arduino para ayudar a identificar paredes aislantes y diatérmicas

es necesario esperar a que se establezca la marcación del sensor en la pantalla LCD para considerar correcta la medición.

3. Para registrar la actividad con más de detalle, los alumnos pueden construir una tabla (Tabla 1) en la que se registra cada tres minutos la temperatura de cada sistema, además de la condición en la que se encuentra el cubo de hielo.

Tabla 1. Tabla propuesta para el registro de datos.

Material	Agua existente en el sistema			Temperatura en °C (tiempo en minutos)									
	Nada	Poca	Mucha	3	6	9	12	15	18	21	24	27	30
Estambre													
Plástico													
Papel aluminio													

RESULTADOS

GUÍA PARA EL PROFESOR: Como la temperatura de cada uno de los sistemas va a ser diferente (aunque no por mucho, uno o dos grados centígrados), esto ayudará a los estudiantes a reflexionar sobre el tipo de material que rodea al cubo de hielo, ya que este se mantendrá durante más tiempo o no en su estado sólido.

El programa que se va a cargar en el Arduino, presenta en la pantalla LCD cuatro escalas, la Celsius, la Kelvin, la Rankin y la Fahrenheit.

ANÁLISIS DE RESULTADOS

GUÍA PARA EL PROFESOR: Como ya se ha mencionado varias veces, los alumnos deben darse cuenta y reflexionar que dependiendo del material que rodee al hielo, será el estado en el que se conserve. Para llegar a esta reflexión se pueden insertar durante el desarrollo de la práctica preguntas como las siguientes:

- ¿El estado de cada cubo de hielo es el mismo? ¿Tienen el mismo tamaño? ¿Sí o no? ¿Por qué?

10. Midiendo la temperatura empleando el Arduino para ayudar a identificar paredes aislantes y diatérmicas

- ¿Sucederá lo mismo con el atole que nos venden en un puesto de tamales, y por eso nos lo dan en un vaso de unicel?; es decir, ¿éste se mantiene durante más tiempo ‘calientito’ en dicho vaso que si lo ponemos en uno de plástico? ¿Sí o no? ¿Por qué?
- Pero que me queme la mano con la que sostengo el vaso, nada tiene que ver con el material del vaso que contiene al atole, ¿sí o no?
- ¿Por qué las personas que viven en la calle cubren su cuerpo con papel periódico?, o bien, ¿por qué al dormir lo hacen sobre cartón?
- ¿Cómo funciona una hielera de unicel?

DISCUSIÓN Y CONCLUSIONES

GUÍA PARA EL PROFESOR: Se espera que los estudiantes distingan la diferencia entre los tipos de paredes que existen y aunque posiblemente no van a mencionar las palabras diatérmica y/o adiabática, seguramente mencionarán la palabra “aislante” y de ahí se puede llegar a las otras dos. Con lo sucedido en la actividad experimental, más las respuestas obtenidas en el análisis de resultados, se esperaría que los alumnos llegaran a la conclusión de que existen materiales que pueden ser muy delgados pero buenos aislantes, o gruesos y muy malos aislantes; es decir, que hasta cierto punto, más que lo grueso o lo delgado de un pared, importa el material del que está hecha. Si con lo expuesto hasta estos momentos no es suficiente quizás el siguiente par de situaciones podría servir:

- ¿Por qué a veces, llegamos a oír comentarios como el siguiente?, “esta chamarra o abrigo está bien grueso y no me tapa nada”.
- Esta chamarra es bien “calientita” y está bien delgada, o es bien práctica porque no es estorbosa.

REFERENCIAS

Loredo A. (2012). Medición de temperatura con Arduino+LM35+LCD2X16. Recuperado el 19 de abril del 2017 de <https://www.youtube.com/watch?v=c6Omj78oojo>

ANEXOS:

1. PROGRAMA:

El programa cargado en el Arduino presenta en la LCD la temperatura en C, K, F y R. En el video, el Dr. explica como hizo la conversión de mV a C y como convirtió los C en R, K y F

/* Programa: Medición de temperatura con Arduino+LM35+LCD2X16

ver video e información en youtube:

https://www.youtube.com/watch?v=c6Omj78oojo&feature=gp-n-y&google_comment_id=z12jynujmxejfx1sk23hcnsxozracv00g

suscríbete a mi canal: <https://www.youtube.com/user/rubyck71>

Profesor: Rubén Loredo Amaro, Universidad Tecnológica de Altamira*/

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11,5,4,3,2);
```

```
float centi()
{
  int dato;
  float c;
  dato=analogRead(A0);
  c = (500.0 * dato)/1023;
  return(c);
}
```

```
float kelvin(float cent)
{
  float k;
  k=cent+273.15;
  return(k);
}
```

```
float fahren(float cent)
{
  float f;
  f = cent*1.8 + 32;
  return f;
}
```

```
float rankin(float cent)
{
  float r;
  r=(cent+273.15)*1.8;
  return r;
}
```

10. Midiendo la temperatura empleando el Arduino para ayudar a identificar paredes aislantes y diatérmicas

```
void setup()
{
  lcd.begin(16,2);
  lcd.print("C=  F=");
  lcd.setCursor(0,1);
  lcd.print("K=  R=");
}

void loop()
{
  float Centigrados = centi();
  float Farenheit = fahren(Centigrados);
  float Kelvin = kelvin(Centigrados);
  float Rankin = rankin(Centigrados);

  lcd.setCursor(2,0);
  lcd.print(Centigrados);

  lcd.setCursor(10,0);
  lcd.print(Farenheit);

  lcd.setCursor(2,1);
  lcd.print(Kelvin);

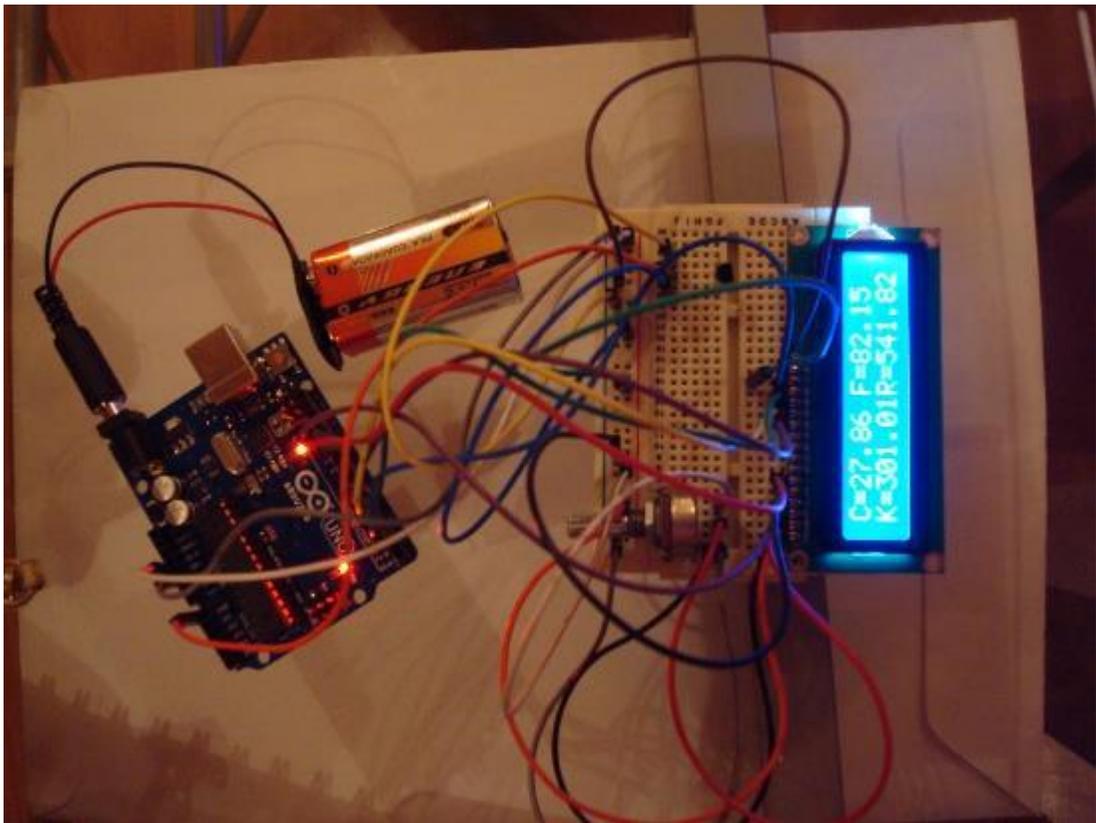
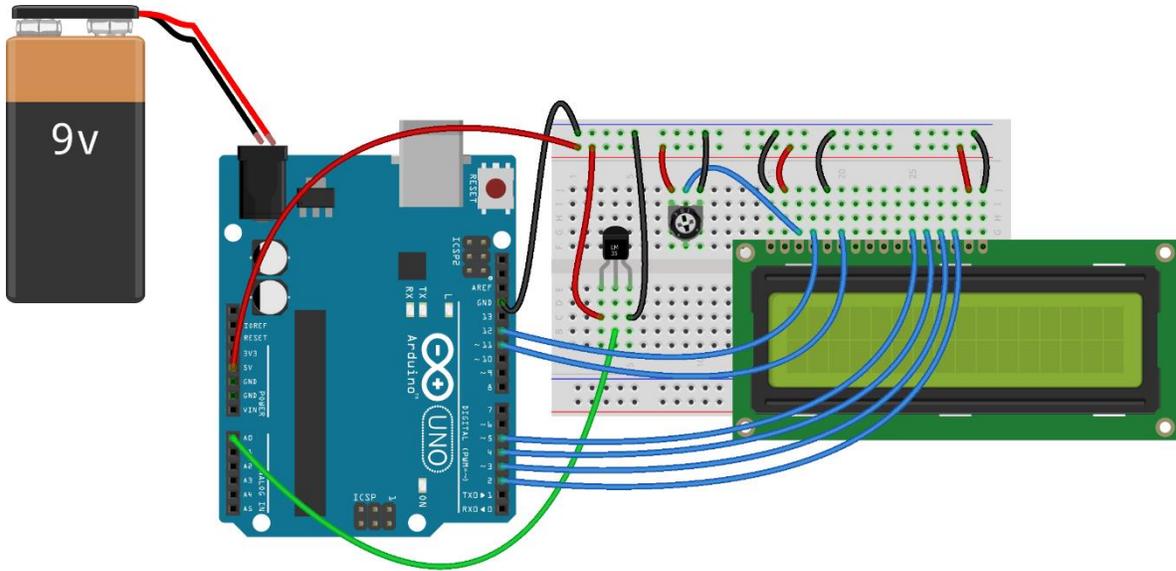
  lcd.setCursor(10,1);
  lcd.print(Rankin);

  delay(200);
}
```

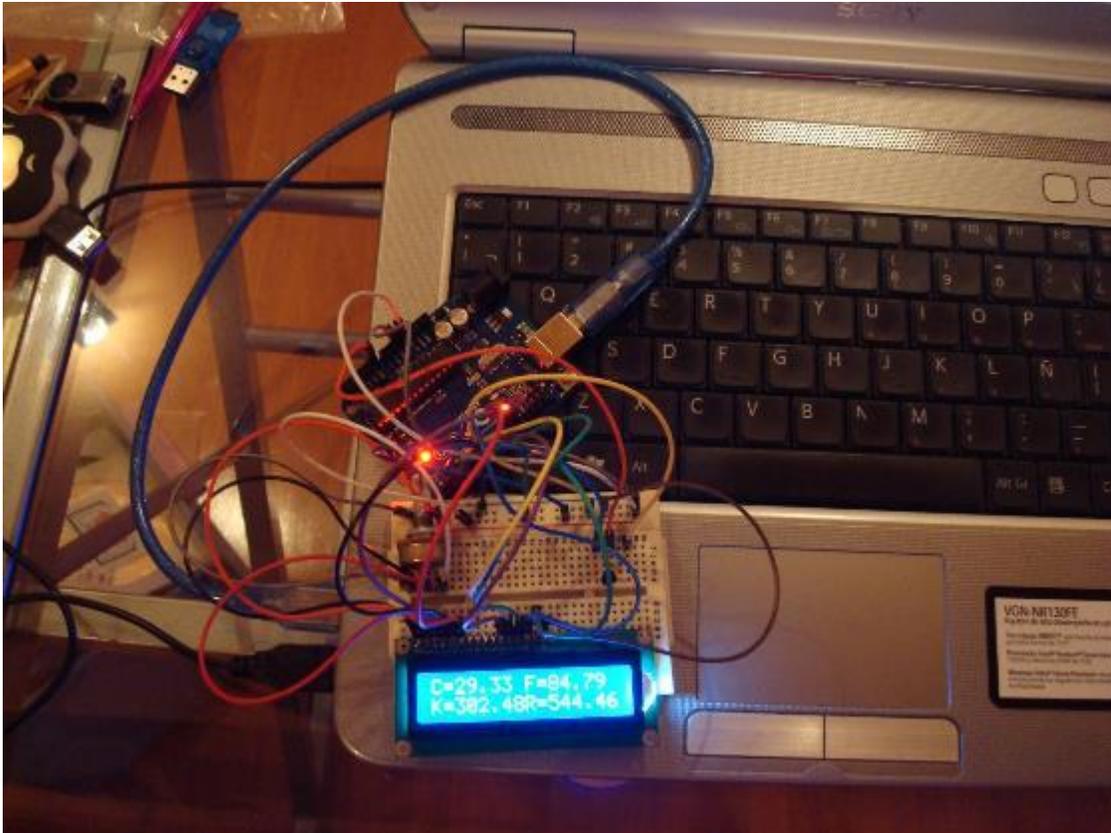
2. DIAGRAMAS

Nota: Aunque el sensor de temperatura aparece completamente incorporado al protoboard, se le deben de colocar unas extensiones con ayuda de unos pedazos de alambre, como unas extensiones, de lo contrario, no es posible meter todo el protoboard al sistema en el que se desea medir la temperatura.

10. Midiendo la temperatura empleando el Arduino para ayudar a identificar paredes aislantes y diatérmicas



10. Midiendo la temperatura empleando el Arduino para ayudar a identificar paredes aislantes y diatérmicas



3. TABLA DE COLORES DE LAS RESISTENCIAS

Color	1ra. banda	2da. banda	3ra. banda	Tolerancia
Negro	0	0	$\times 10^0$	
Café	1	1	$\times 10^1$	1 %
Rojo	2	2	$\times 10^2$	2 %
Naranja	3	3	$\times 10^3$	
Amarillo	4	4	$\times 10^4$	
Verde	5	5	$\times 10^5$	
Azul	6	6	$\times 10^6$	
Violeta	7	7	$\times 10^7$	
Gris	8	8	$\times 10^8$	
Blanco	9	9	$\times 10^9$	
				Dorado 5%
				Plata 10%

11. SENSOR ULTRASÓNICO PARA INVIDENTES

Responsable: David León Salinas ENP 7	Participantes: Bernal González Josué Castellanos Hernández Lesly Betsy Garibaldi Arriaga Jorge Juan Francisco Maya	Revisión: Eduardo José Vega Murguía ICAT
---	--	--

Ubicación curricular

Escuela Nacional Preparatoria

Física IV Área 2

Unidad 1. Física de la visión y la audición

INTRODUCCIÓN

El empleo de conceptos de Física con la tecnología, nos ayuda a diseñar nuevos dispositivos capaces de ayudar a personas invidentes, y mejorar su calidad de vida al prevenir y orientarlos en su transitar por la casa, calle o espacios públicos. Cuando uno va por la calle puede percatarse de que está llena de obstáculos, que afectan principalmente a la gente incapacitada para ver en su camino (invidente). El proyecto Dispositivo Ultrasónico Blind consiste en un dispositivo que se coloca en la cabeza, como una lámpara de minero, con un sensor ultrasónico en la parte frontal, el cual emite ondas sonoras. Cuenta con un microcontrolador Arduino UNO que mide de una forma interna el tiempo que toma para que las ondas ultrasónicas regresen.

OBJETIVOS

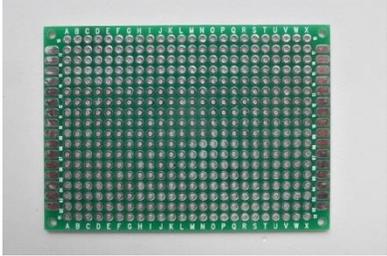
Investigar sobre el sensor ultrasónico de Arduino UNO.

Diseñar un dispositivo portátil con la finalidad de ayudar a las personas invidentes a orientarse en su entorno a través del sonido.

MATERIAL

Descripción	Imagen	Comentarios
Tarjeta Arduino UNO		El hardware consiste en una placa de circuito impreso con un microcontrolador, usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (shields) que amplían las características de funcionamiento de la placa Arduino.

11. Sensor ultrasónico para invidentes

Descripción	Imagen	Comentarios
Cable USB A macho a B macho		Para establecer la comunicación entre la tarjeta Arduino y el programa.
Sensor ultrasónico HCSR04		Pines de conexión: 1.-VCC 2.-Trig (<i>Disparo del ultrasonido</i>) 3.-Echo (<i>Recepción del ultrasonido</i>) 4.-GND
Placa fenólica o tarjeta protoboard.		Componente para hacer las conexiones entre el sensor ultrasónico HCSR04 y el microcontrolador Arduino.
Bocina de 8 Ω		Dispositivo que emitirá sonido al mandar una señal el sensor ultrasónico
Lámpara de minero		Es el espacio donde se alojará el dispositivo

11. Sensor ultrasónico para invidentes

Descripción	Imagen	Comentarios
Batería de 9V y adaptador (broche 9 V a plug invertido)		Es la principal fuente de energía externa de nuestro dispositivo.
Cables de conexión. Jumper macho-macho		

Software

- Programa Arduino IDE 1.8.2 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>.
- Programa que se debe grabar en Arduino para hacer el experimento.

DESARROLLO

1.- Realizar las conexiones de la placa Arduino con el sensor, en los pines de salida digitales (13,12,8), alimentación (5 V) y tierra (GND). El diagrama se presenta en el Anexo.

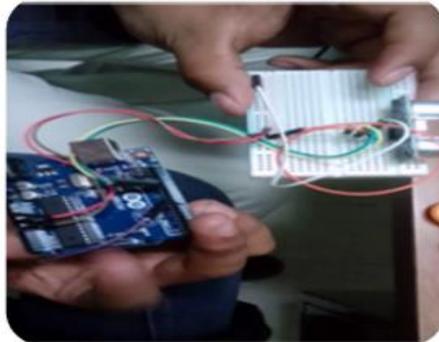


Figura 1. Verificación de las conexiones.

2.- Después conectar la bocina a la salida digital 8 y tierra de la Protoboard.

11. Sensor ultrasónico para invidentes

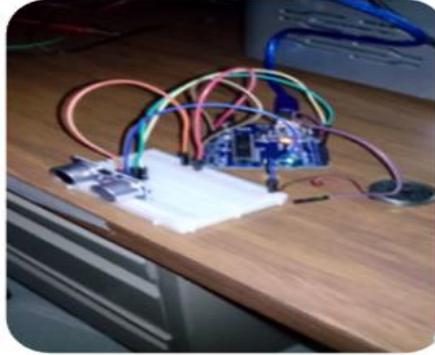


Figura 2. Montaje de la placa.

3.- Hacer la programación y subirla al microcontrolador de la placa Arduino por el puerto USB.



Figura 3. Programación del dispositivo.

4.- Hacer comprobación de la programación midiendo con una regla y un objeto si es correcto que a 60 cm la bocina hace un *beep* (pulso acústico), y a 30 cm un doble *beep* acelerado.



Figura 4. Mediciones de la programación.

5.- Una vez que el dispositivo funciona adecuadamente, se procede a montarlo en una lámpara de minero, sacando todo lo que la compone.

11. Sensor ultrasónico para invidentes



Figura 5. Soldando el prototipo.

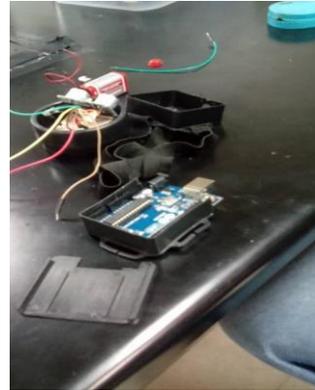


Figura 6. Adaptación del prototipo.

6.- Una vez ya montado comenzar las pruebas y reajustes para las necesidades de la persona invidente el cual se entera por el bip que emite la bocina.



Figura 7. Probando el dispositivo.



Figura 8. Integrantes del equipo.



Figura 9. Alumno invidente probando el dispositivo.

RESULTADOS

El desarrollo del dispositivo cumplió con nuestro objetivo, ya que se construyó un novedoso prototipo que ayudará a las personas invidentes a orientarse y tener una noción de la distancia.

La implementación del dispositivo en la cabeza nos ayudó a aprovechar la rotación de esta misma, y así evitar el uso de varios sensores. La eficacia del dispositivo dependerá de la densidad y forma de la superficie, debido que el sonido es absorbido y rebotado de distinta manera por cada superficie y

11. Sensor ultrasónico para invidentes

medio. Al probar el dispositivo en nuestro compañero invidente, él disminuyó el número de choques al día y tuvo más seguridad en su desplazamiento por la escuela.



Figura10 Pruebas caminando en la prepa.

Se puede consultar una demostración del funcionamiento del sistema en las siguientes ligas a videos (recuperados el 4 de junio de 2019):

<https://www.youtube.com/watch?v=DMczzMM4DsI>

<https://www.youtube.com/watch?v=q13LlnxWmuc>

<https://www.youtube.com/watch?v=c8FO4nx70XE>

<https://www.youtube.com/watch?v=1Pex7jYGoLY>

DISCUSIÓN Y CONCLUSIONES

Consideramos que uno de los retos de un alumno de Física de área 2, es lograr entender un concepto físico y relacionarlo con una aplicación práctica. En este caso se ha utilizado el microcontrolador Arduino UNO y el sensor ultrasónico HCSR04, con la finalidad de diseñar un instrumento que logra orientar a un invidente al medir la distancia a la cual se encuentra un obstáculo.

CONCLUSIONES

Los objetivos fueron cumplidos, sin embargo, afirmamos que el dispositivo ultrasónico Blind no va a sustituir el bastón sino que lo va a complementar y brindará otras garantías de seguridad. El Dispositivo Ultrasónico Blind es una alternativa para que las personas invidentes se apoyen a través de sonido, ya que les proporciona información de la presencia de un obstáculo próximo, al desplazarse en un lugar que no conozcan.

El hecho de aplicar un concepto de física a un proyecto es un acto de servicio social, ya que a través de la tecnología y la física se pudo realizar un dispositivo que utiliza el ultrasonido para mejorar la calidad de vida de las personas, y evitarles lesiones, principalmente en la cabeza. También se le dio

11. Sensor ultrasónico para invidentes

una noción de distancia con base al sonido de la bocina, ya que el alumno escucha un sonido cuando el obstáculo está a 60 cm, y un sonido más intenso cuando está a 30 cm.

REFERENCIAS

Crawford Jr. 2007. (1977). Ondas. Berkeley Physics Course. Volumen III. España: Editorial Reverté.

Giancoli (1985) FÍSICA. Principios y Aplicaciones. España: Editorial Reverté.

Hewitt, Paul G. (2004). Física Conceptual, Novena Edición. México: Pearson.

Serway, R. (2007). Física, conceptos y aplicaciones. Quinta edición. México: McGraw-Hill.

11. Sensor ultrasónico para invidentes

ANEXOS

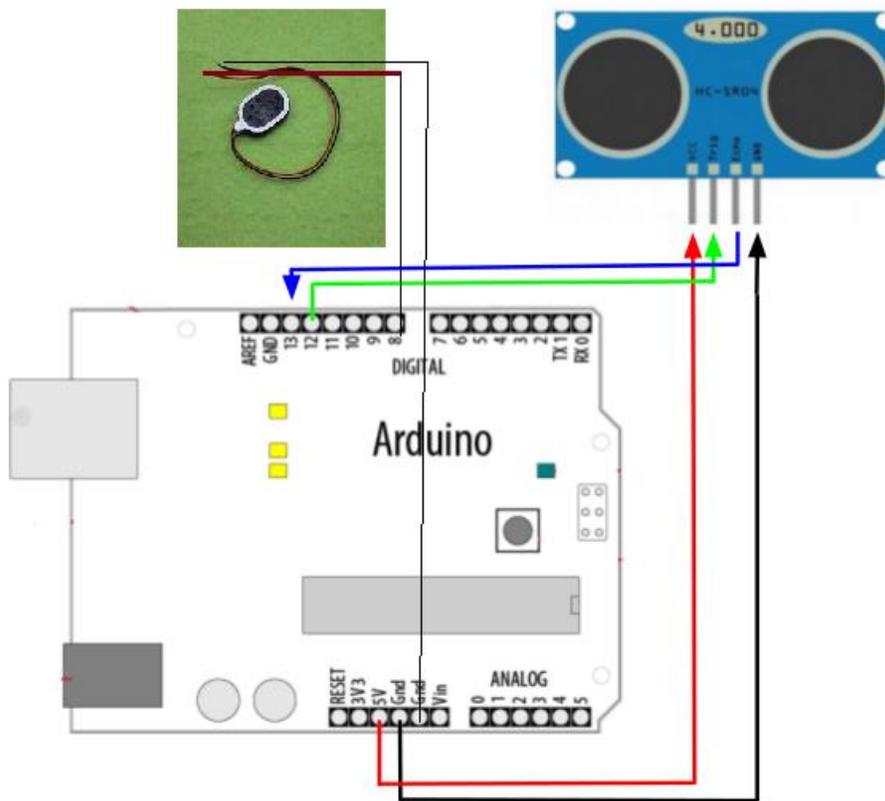
1.- Listado completo del programa que se debe grabar en Arduino para hacer el experimento

```
//Definición de los pines de conexión del sensor ultrasónico
#define trigPin 12
#define echoPin 13
// trig es para disparar la señal y echo es un pulso que indica la señal de eco
void setup() {
  Serial.begin (115200); // Se utiliza la máxima velocidad disponible
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
//Funciones para diferentes velocidades de pulso acústico (beep)
void beepFast(){
  // produce una nota en el pin 8 por 200 ms
  tone(8, 440, 200);
  delay(50);
  // Apaga la nota
  noTone(8);
  delay(300);
}
void beepSlow(){
  tone(8, 440, 200);
  delay(500);
  noTone(8);
  delay(300);
}
void beepMedium(){
  tone(8, 440, 200);
  delay(200);
  noTone(8);
  delay(300);
}
//No produce beep
void beepNo(){
  noTone(8);
  delay(300);
}
// Envía un pulso y calcula el tiempo de que tarda en llegar el eco
void loop() {
  int duration, distance;
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(1000);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  // Verifica que la distancia y el beep estén dentro del intervalo de operación
  if (distance >= 200 || distance <= 0){
```

11. Sensor ultrasónico para invidentes

```
Serial.println("Fuera de intervalo");  
}  
else if (distance >=0 && distance <=30){  
  beepFast();  
  Serial.print(distance);  
  Serial.println(" cm");  
}  
else if (distance >=31 && distance <=60) {  
  Serial.print(distance);  
  Serial.println(" cm");  
  beepMedium();  
}  
}
```

2.- Diagrama esquemático del circuito electrónico que se conecta al Arduino



12. LEY DE BOYLE

Responsable: Milagros Pacheco Castañeda ENP 5	Participantes: Miguel Á. Bañuelos Saucedo ICAT	Revisión: Eduardo José Vega Murguía ICAT
---	--	--

Ubicación curricular

Escuela Nacional Preparatoria

Informática aplicada a la Ciencia y a la Industria

Unidad 2. Análisis de los datos e interpretación de resultados

Unidad 3. Automatización y control de procesos

INTRODUCCIÓN

Cuando ocurre un proceso en el cual la temperatura de un sistema se mantiene constante, se dice que tenemos un proceso isotérmico. En 1643, el científico italiano Evangelista Torricelli (1608-1674), demostró que una columna de gas podía ejercer presión y que ésta podía medirse. Este trabajo atrajo la atención del químico inglés Robert Boyle (1627-1691), quién fue la primera persona que estudió este tipo de procesos. En 1660 demostró que el volumen de un gas es inversamente proporcional a la presión, es decir, que al duplicar el volumen la presión disminuye a la mitad de su valor inicial. En 1662 reportó los resultados de sus experimentos llegando a la conclusión de que “el volumen de una cantidad fija de un gas a temperatura constante es inversamente proporcional a la presión del gas”. Conocida como la ley de Boyle y puede expresarse matemáticamente como:

$$V \propto \frac{1}{P}$$

Donde V y P son respectivamente el volumen y la presión del gas.

Una forma de entender la ley de Boyle es tener un sistema semejante a una jeringa con émbolo, en la cual se tiene una cantidad fija de un gas a determinadas condiciones de presión, temperatura y volumen (ver figura 1).

12. Ley de Boyle

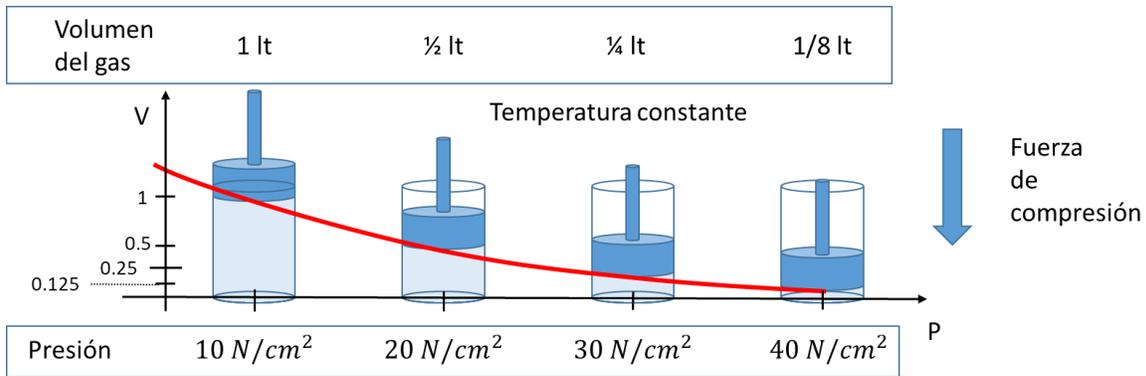


Figura 1. Ley de Boyle. La imagen muestra en la gráfica cómo el volumen en litros de un gas disminuye conforme aumenta la presión que recibe.

Señales Analógicas y digitales

En la naturaleza, existen manifestaciones de un suceso o pequeñas cantidades de una cosa, el percibir esas manifestaciones es lo que llamamos señales. Para la Física, una señal es una variación de la corriente eléctrica, o de otra magnitud, que se utiliza para transmitir la información.

Una señal analógica, es un tipo de señal que se interpreta mediante una función matemática continua, representando un dato de información en función del tiempo.

Las señales analógicas como el sonido, la luz o la energía, se pueden percibir en la naturaleza en todos los lugares, son señales que tienen una variación continua, donde la información para pasar de un valor a otro pasa por todos los valores intermedios; es decir, puede tomar infinitos valores.

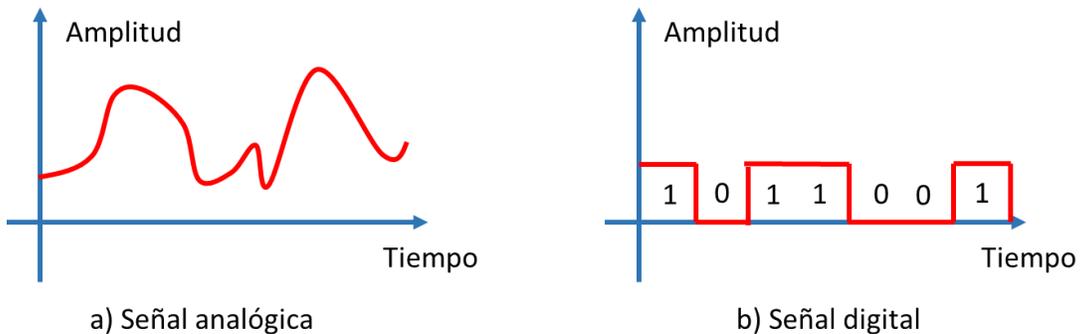


Figura 2. Señales analógicas y digitales. A la izquierda: una señal analógica (toma cualquier valor). A la derecha: una señal digital (toma uno de dos valores).

La señal digital es aquella que presenta una variación discontinua con el tiempo y que sólo puede tomar ciertos valores discretos (adquiere valores de un conjunto, no cualquier valor).

Las señales digitales no se producen en el mundo físico de forma natural como tales, sino que son creadas por el hombre, y sólo pueden tomar dos valores o estados: 0 y 1, que pueden ser impulsos eléctricos de baja y alta tensión, interruptores abiertos o cerrados, etc. La señal digital se puede originar mediante el cambio discreto de las mismas magnitudes que caracterizan la señal analógica. Una señal eléctrica que presenta saltos de voltaje entre 0 y 5 voltios es una señal digital.

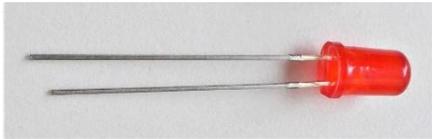
12. Ley de Boyle

OBJETIVOS

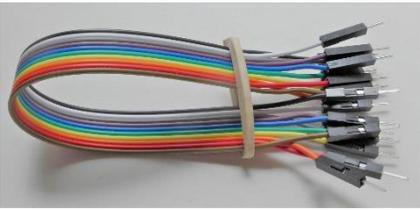
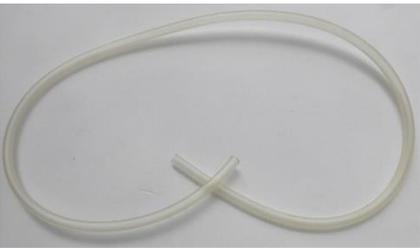
El alumno:

- Deducirá la relación que existe entre la presión y el volumen de un gas a través de actividades experimentales para explicar la ley de Boyle de los gases.
- Practicará procesos de adquisición de datos con el uso de herramientas digitales realizando mediciones de magnitudes analógicas.
- Analizará e interpretará los datos obtenidos y realizará gráficas en hoja electrónica con todos sus elementos (variables y unidades).

MATERIAL

Descripción	Imagen	Comentarios
Resistencia 220 ohms a 1/2 W		Se va a emplear una resistencia de 220 ohms, bandas rojo, rojo, café
LED		Recordar que el cátodo o pata más pequeña es el negativo y la más larga es el ánodo o positivo.
Sensor MPX5700DP		El sensor de presión será el encargado de hacer la lectura analógica. El pin con una pequeña muesca es el pin número uno. Al sensor se le conecta una manguerita de 5 mm de grueso (como las que se utilizan en los acuarios).
Arduino UNO		

12. Ley de Boyle

Descripción	Imagen	Comentarios
Cable USB A macho a B macho		
7 Cables jumpers macho-macho		Se utilizan para transferir las señales eléctricas de la placa de prototipos a los pines de entrada/salida del Arduino.
Jeringa 35 ml		Se sugiere que la jeringa sea de 35 ml. para que un bombón o un globo pueda introducirse dentro. Si se encuentra una jeringa más grande mejor
Un trozo de manguera para pecera		
Globo o un bombón pequeño		Los globos deben ser del número 0 más pequeño. Los bombones son los más pequeños.

Software

- Google Drive Apps. Disponible en: www.google.com. Documentos. Hoja de cálculo.
- Programa Arduino IDE 1.8.2 o superior. Disponible en: <https://www.arduino.cc/en/Main/Software>.
- Programa CoolTerm 1.4.7 o superior. Disponible en: <http://freeware.the-meiers.org>.
- Hoja de Cálculo (Excel).

GUÍA PARA EL PROFESOR: Se sugiere solicitar a los alumnos crear una cuenta de correo en Gmail para acceder a las apps de GoogleDrive.

12. Ley de Boyle

DESARROLLO

Actividad 1. Introducción a los gases

1. Formar equipos de trabajo de 2 participantes.
2. Crear un documento en Google Drive que se convertirá en el reporte de tu práctica con el nombre: *Ley de Boyle*.
3. Realizar una breve investigación acerca de los postulados de la ley de Boyle.
4. Ilustrar cada postulado, revisando tanto fuentes impresas como electrónicas y registrarlas en formato APA.

Actividad 2. Señales analógicas y digitales

5. Observar el siguiente video: Club de jóvenes por la ciencia (). *Señales analógicas y digitales*. Recuperado el 8 febrero de 2016 de https://www.youtube.com/watch?v=tn3W_lvdk9w&t=3s
6. Realizar una investigación acerca de las diferencias entre señales analógicas y digitales tanto en fuentes impresas como electrónicas.
7. Realizar en la práctica (Documento de Google) una tabla comparativa con el título
 - a. “Señales digitales vs Señales analógicas”

Actividad 3. Práctica de Arduino con sensor de presión

8. Utilizando el material para la práctica, construir el siguiente circuito. (Anexo 1)

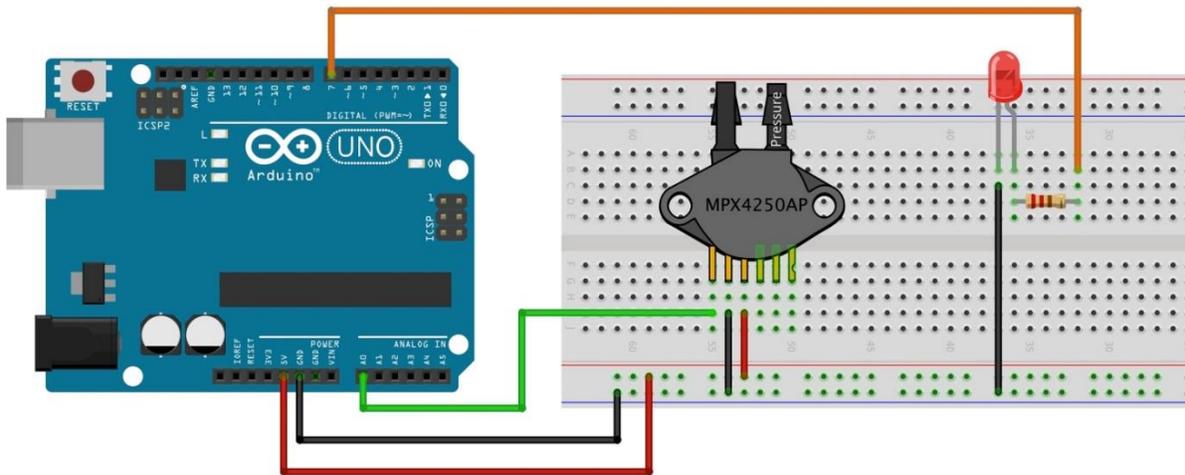


Figura 3. Diagrama de conexiones de un circuito con sensor MPX5700 y LED.

9. Una vez concluido el circuito, conectar la jeringa de la siguiente forma:
10. Quitar el émbolo a la jeringa e insertar dentro un bombón o un globo inflado del tamaño de la jeringa. Volver a colocar el émbolo y observar el volumen que ocupa el globo/bombón.

12. Ley de Boyle

11. Conectar el extremo del sensor de presión a la jeringa.

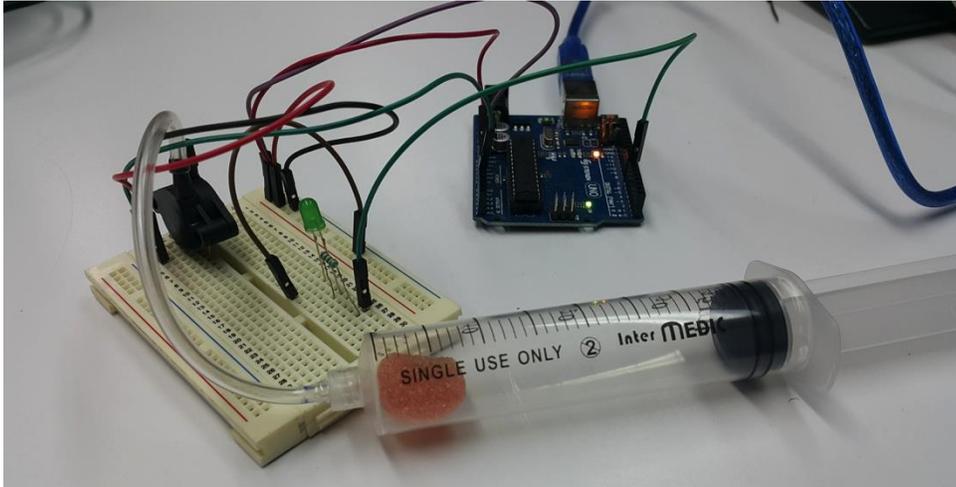


Figura 4. Circuito completo para experimento.

12. Empujar el émbolo de la jeringa lo más posible, observar y responder en la práctica:

- ¿Qué le pasa al globo/bombón cuando se empuja el émbolo?
- ¿Qué sucede con el volumen del gas dentro del globo/bombón cuando se aplica más fuerza al émbolo para comprimir el aire y aumentar la presión?

13. Jalar el émbolo hacia fuera y contestar:

- ¿Qué sucede con el volumen del gas contenido en el Globo/bombón cuando se jala el émbolo?
- ¿Qué cambio ocurre en la presión al jalar el émbolo hacia afuera? ¿Por qué?

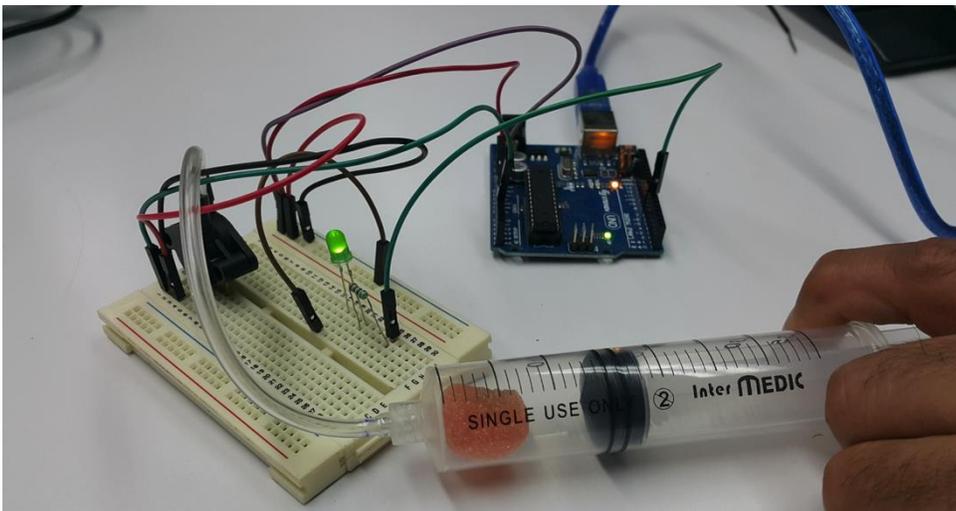


Figura 5. Empujando el émbolo de la jeringa.

14. Con la ayuda del compañero de equipo, fotografiar cada fase del experimento para ilustrarlo.

12. Ley de Boyle

Registro de presión

15. Iniciar el programa Arduino y capturar el programa para manejar el sensor de presión (Anexo 2) en un sketch nuevo.
16. Analizar las instrucciones del código y contestar en la práctica lo siguiente:
 - a. ¿Con qué instrucción se lee un valor analógico?
 - b. ¿Qué dispositivo electrónico realiza la función de medir la señal analógica?
 - c. ¿En qué momento el valor se convierte de analógico a digital?
 - d. ¿Qué instrucciones se utilizan para convertir a una señal digital?

A continuación, se realizará el mismo experimento, pero registrando los valores que se obtienen con la ayuda del sensor de presión y el monitor serial.

17. Activar la función del monitor serial, ubicada dentro de la barra de funciones (ver figura 6).



Figura 6. Activación del menú serie dentro del menú Herramientas.

Registro de cada medición proporcionada:

18. Colocar el émbolo en el volumen de 35 ml. de la jeringa y registrar el valor de presión que muestra la pantalla serial.
19. Colocar ahora el émbolo en 30 ml, presionar el émbolo y registrar el valor de presión correspondiente que muestra la pantalla serial.

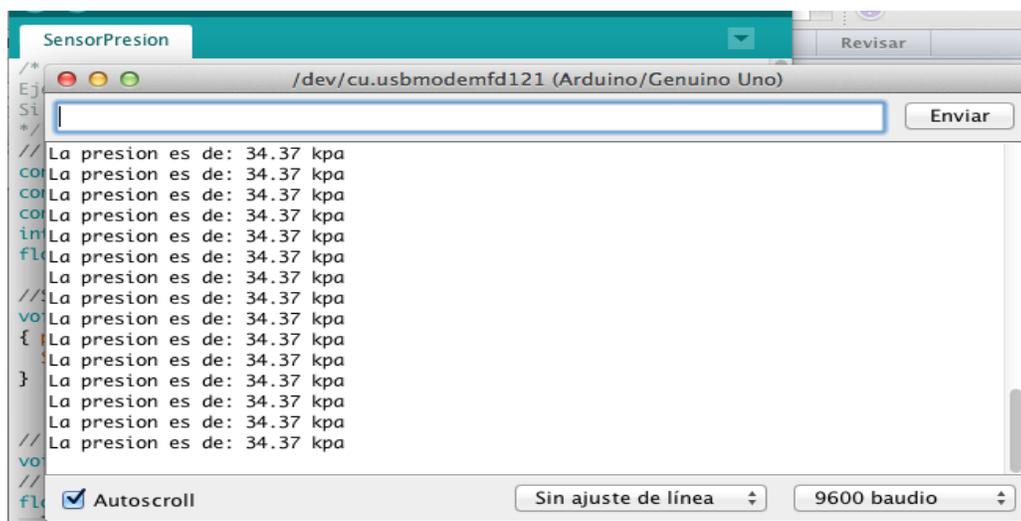


Figura 7. Ventana del monitor serial desplegando los datos de la temperatura.

12. Ley de Boyle

20. Continuar así hasta llegar a 5 ml. Se puede utilizar la tabla 1 para registrar los valores que muestra la pantalla serial.

Tabla 1. Registro de Presión vs Volumen

Volumen ml	Presión KPa
35	
30	
25	
20	
15	
10	
5	

21. Abrir una hoja de Excel, capturar los datos y realizar una gráfica con los valores obtenidos. Las variables son Volumen contra Presión.
22. Observar los datos obtenidos en el experimento y la forma de la gráfica, tomando en cuenta los postulados de la ley de Boyle, contestar:
- ¿Cuál es la relación que guarda la presión con el volumen? ¿Por qué?
 - ¿Qué propiedades se mantuvieron constantes durante el experimento?
 - Reflexionar y redactar una conclusión sobre lo aprendido.
 - Ilustrar la respuesta y conclusiones con fotografías del experimento.

Actividad 4. Graficación

Para la obtención de datos de forma automática y poder realizar las gráficas, se va a utilizar el programa CoolTerm para guardar los datos capturados en un archivo.

- Descargar el programa CoolTerm e instalar (liga para descargar : <http://Freeware.the-meiers.org>).
- Modificar el programa principal eliminando o comentando los textos que se envían con la instrucción Serial.print, sustituyendo la instrucción que envía el valor de la presión por Serial.println. (Anexo 3)
- Ejecutar el programa CoolTerm.
- Seleccionar en el menú la opción Connection → Connect (ver figura 8).

12. Ley de Boyle

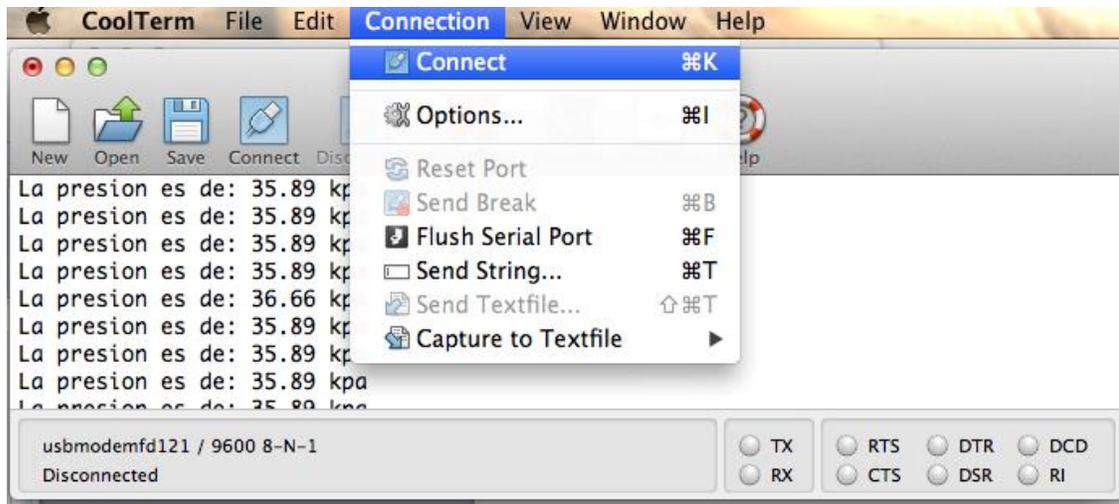


Figura 8. Conexión del programa CoolTerm.

27. Para registrar los datos en un archivo, activar la opción de registro en el menú Connection->Capture to Textfile (ver figura 9).

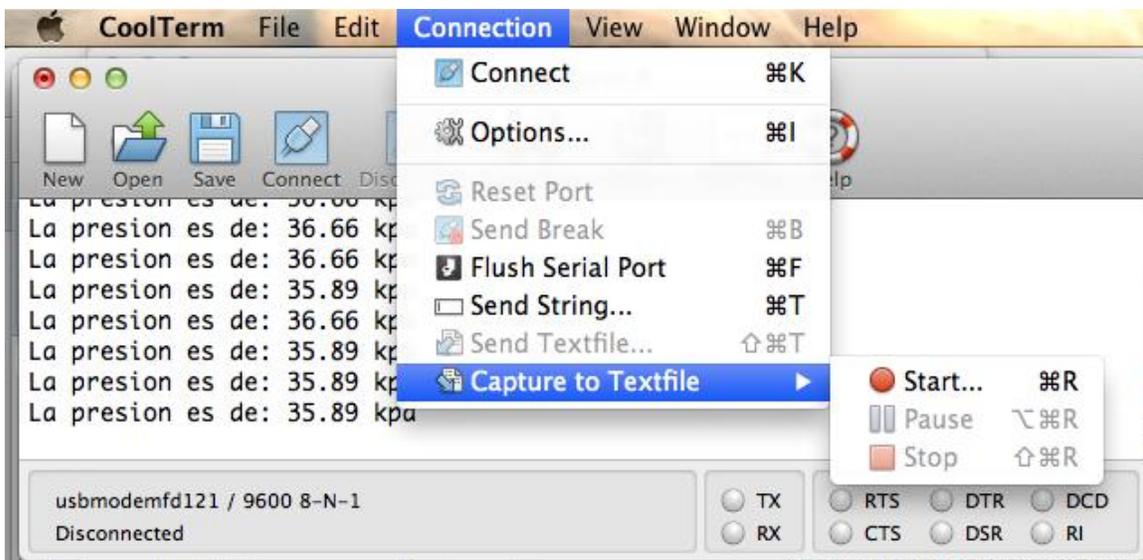


Figura 9. Activación del registro de datos en el programa CoolTerm.

28. Salvar el archivo con el registro de los datos (ver figura 10).

12. Ley de Boyle

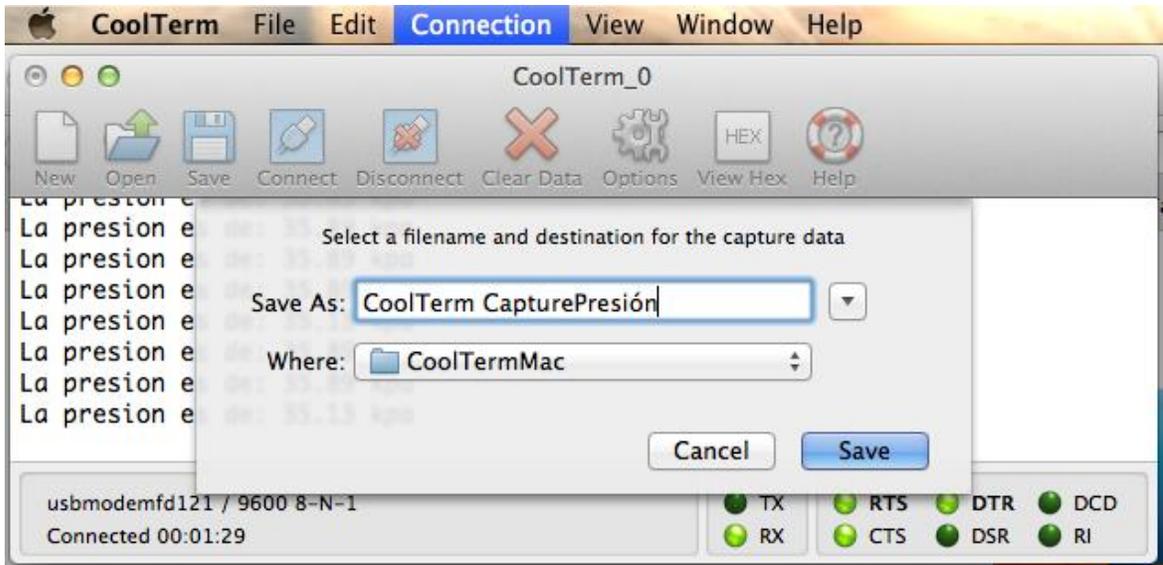


Figura 10. Guardar el archivo con el registro de datos.

29. Ejecutar la hoja de cálculo Excel.

30. Importar el archivo con el registro de datos.

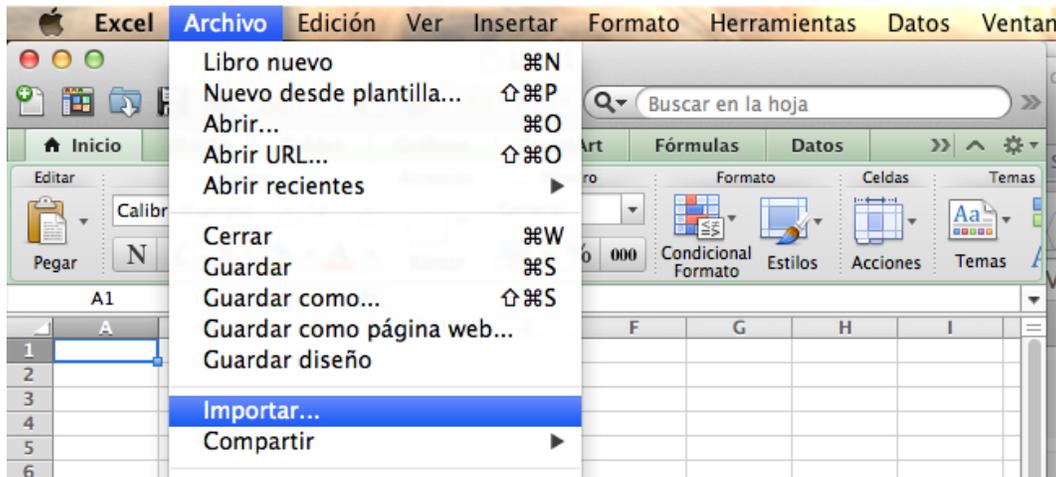


Figura 11. Opción Importar del programa Excel.

31. Del menú de opciones que se despliegan, seleccionar la opción *Archivo de texto* y enseguida escoger el archivo que contiene los datos registrados.

32. Seguir los pasos del asistente para importar datos de Excel, seleccionando la opción *De ancho fijo* del paso 1 de 3.

33. En el paso 2 de 3, establecer el ancho de columna marcando el principio y el fin de la columna con los valores a graficar (figura 12). Una vez concluido presionar la opción Finalizar.

12. Ley de Boyle

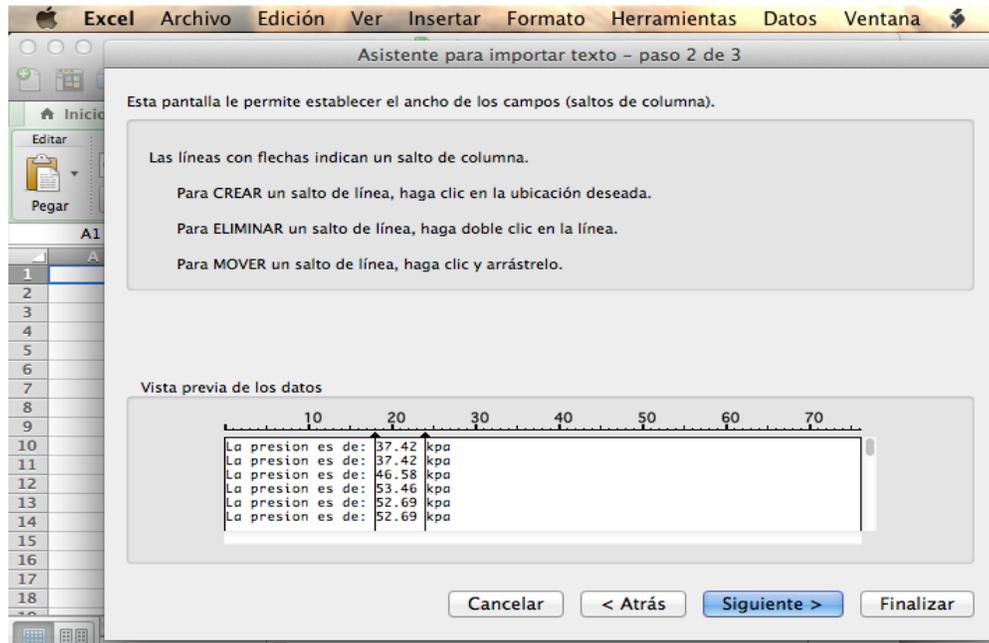


Figura 12. Inicio y fin de las columnas con el registro de datos.

34. Tomar los valores de presión obtenidos y proceder a realizar la gráfica de presión contra tiempo.
35. En Excel seleccionar las columnas a graficar y en el menú seleccionar Insertar->Gráfico-líneas (ver figura 13).

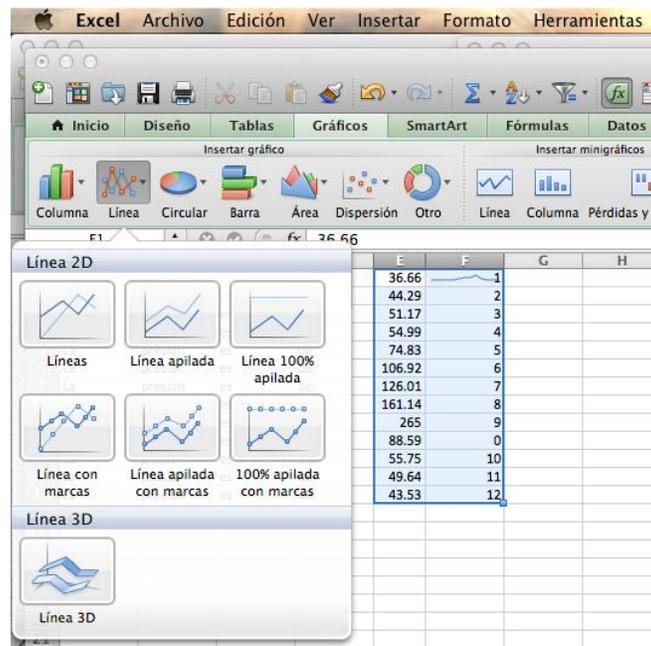


Figura 13. Selección de rangos de datos y tipo de gráfica en Excel.

12. Ley de Boyle

36. Insertar las gráficas en la práctica y redactar una conclusión de lo obtenido.

RESULTADOS

En la figura 14 se muestra un ejemplo de gráfica completa esperada, Presión vs. tiempo.

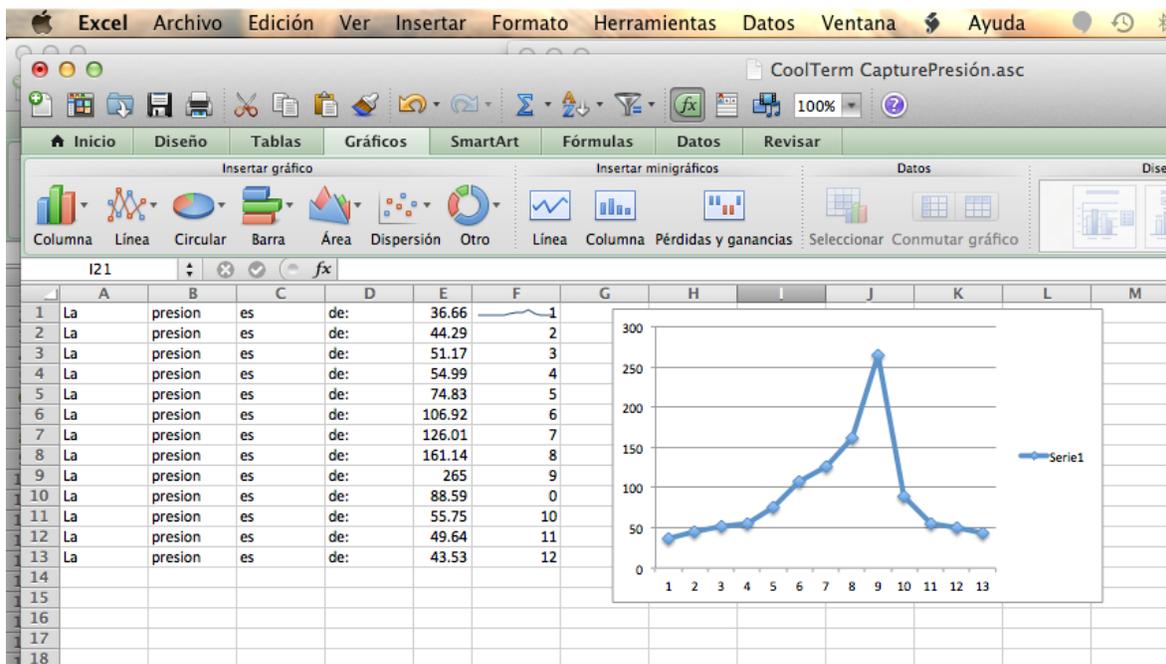


Figura 14. Ejemplo de tabla y gráfica Presión vs. Tiempo.

DISCUSIÓN Y CONCLUSIONES

Actividad 5. Conclusiones

Una vez concluido el experimento, reflexionar y contestar las siguientes preguntas:

- ¿Qué ventajas de las señales digitales permiten realizar este experimento?
- ¿El cambiar la señal analógica a digital, que procedimientos son posibles de hacer con los datos del experimento?
- ¿Qué ventajas se observa al trabajar con las señales digitales en lugar de analógicas?
- Redactar una breve reflexión acerca de las ventajas de manejar señales digitales en el control de experimentos.
- Observar los datos registrados tabla No. 1 y la gráfica correspondiente (Presión vs. Volumen)
 - ¿Cómo es la relación entre el volumen y la presión observada en el experimento?
 - ¿Cómo se muestra la gráfica? ¿Por qué se generó de esa forma?
 - ¿Los postulados de la ley de Boyle permiten explicar lo ocurrido en el experimento? ¿Porqué?
- Leer la página 12 del documento Teoría Cinético Molecular del colegio de Química. (Anexo 4): <https://drive.google.com/file/d/0BjyaWwjdvhdWndLZXcwWUQycHM/view> y responder:
 - ¿Qué relación tiene la ley de Boyle en el buceo?

12. Ley de Boyle

- ¿Por qué es el buceo un deporte de alto riesgo?
- ¿Crees que para bucear es importante conocer las leyes de los gases? ¿Porqué?
- Redacta una breve reflexión acerca de lo aprendido en el desarrollo de la práctica.
- Entregar la práctica con el producto realizado de las siguientes actividades:

Actividad 1. Introducción a los gases

- Investigación

Actividad 2. Señales analógicas y digitales

- Tabla comparativa “Señales digitales vs. Señales analógicas.”

Actividad 3. Práctica de Arduino con sensor de presión

- Desarrollo del experimento

Actividad 4. Graficación

- Gráficas “Volumen vs. Presión” y “Presión vs. tiempo”

Actividad 5. Conclusiones

- Cuestionario contestado con reflexiones y conclusiones

37. Consultar la rúbrica de evaluación (Anexo 4) para verificar los puntos a evaluar.

Conclusiones

Según el experimento realizado con la ayuda del sensor de presión, se puede verificar el postulado de la ley de Boyle, que establece lo siguiente:

1. “La presión de un gas en un recipiente es inversamente proporcional al volumen del recipiente, cuando la temperatura es constante.”
2. El volumen es inversamente proporcional a la presión:
 - Si la presión aumenta, el volumen disminuye.
 - Si la presión disminuye, el volumen aumenta.

REFERENCIAS

Becerra Córdova G. (). Ley de Boyle. Universidad Autónoma de Chapingo. Recuperado el 6 de junio de 2019 de <http://virtual.chapingo.mx/fis/Boyle.pdf>

Club de jóvenes por la ciencia (2016). Señales analógicas y digitales. Recuperado el 6 de junio de 2019 de https://www.youtube.com/watch?v=tn3W_lvdk9w&t=3s

EcuRed (2015). Señales analógicas y digitales. Recuperado el 6 de junio de 2019 de http://www.ecured.cu/index.php/Señales_analógicas_y_digitales

Google (2015). Ayuda de editores de documentos. Recuperado el 5 de noviembre de <https://support.google.com/docs/#topic=2811806>.

Programación y más (2015). Señales analógicas y digitales. Recuperado el 6 de junio de 2019 de: <https://youtu.be/Jp41luUgnyA>

Química. (2015) Teoría Cinética Molecular. DGENP-UNAM. Recuperado el 17 de octubre de 2015 de <https://drive.google.com/a/enp.unam.mx/file/d/0BYjaWwjdvhdWndLZXcwWUQycHM/view>

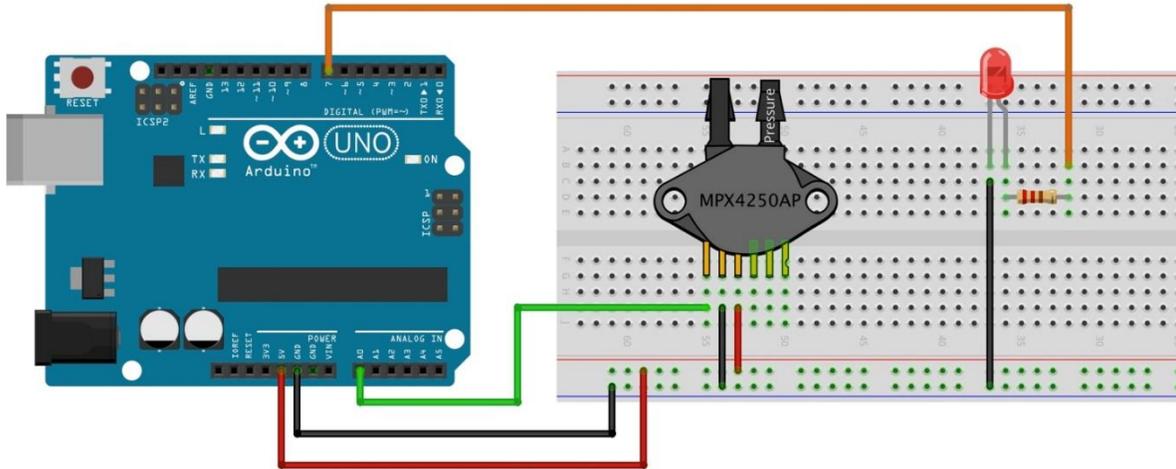
12. Ley de Boyle

UAEH (--) Comunicación en Redes. Apuntes digitales. Recuperado el 17 de octubre de 2016 de <http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro27/index.html>

12. Ley de Boyle

ANEXOS

Anexo 1. Diagrama esquemático del circuito electrónico que se conecta al Arduino.



Anexo 2. Programa para grabar en el Arduino. Ley de Boyle.

```
/*
```

```
Ejemplo con el sensor de presión MPX5700DP  
Si la presión supera un valor umbral se encenderá un LED
```

```
*/
```

```
// En esta sección se definen las constantes  
const int pinSensor = A0; // Entrada analógica. Se conecta al sensor de presión  
const int pinLed = 7; // Pin de salida digital para el LED  
const int umbral = 40; // Umbral de la presión en kilopascales  
int valorSensor = 0; // Variable para guardar el valor leído del sensor  
float presion = 0; // Variable para guardar la presión//
```

```
//Sección de inicialización
```

```
void setup()  
{ pinMode(pinLed, OUTPUT);  
  Serial.begin (9600);  
}
```

```
// Programa principal
```

```
void loop() {  
  float volts; // Se define una variable de punto flotante  
  valorSensor = analogRead(pinSensor); // leemos el valor del sensor
```

12. Ley de Boyle

```
volts = 5*(valorSensor / 1023.0); // Se convierte a volts
presion = (volts*1000)/6.4; // convierto los volts entre 6.4 para que de los kilopascales
if (presion > umbral){ // Se compara la presión con el umbral
    digitalWrite(pinLed, HIGH); // Si se supera el umbral se enciende el LED
}
else {
    digitalWrite(pinLed, LOW); // En caso contrario se apaga el LED
}
Serial.print("La presion es de: "); // Envía un letrero a la PC
Serial.print(presion); // Envía el dato de la presión
Serial.println(" kpa"); // Envía un letrero a la PC
delay(1000); // Se introduce un retardo de 1 segundo antes de repetir el programa
}
```

Anexo 3. Código modificado para utilizar el programa CoolTerm (sustituir el programa principal void loop())

```
// Programa principal
void loop() {
float volts; // Se define una variable de punto flotante
valorSensor = analogRead(pinSensor); // leemos el valor del sensor
volts = 5*(valorSensor / 1023.0); // Se convierte a volts
presion = (volts*1000)/6.4; // convierto los volts entre 6.4 para que de los kilopascales
//Serial.print("La presion es de: "); // Envía un letrero a la PC
Serial.println(presion); // Envía el dato de la temperatura a la PC con el programa CoolTerm
//Serial.println(" kpa"); // Envía un letrero a la PC
delay(1000); // Se introduce un retardo de 1 segundo antes de repetir el programa
}
```

12. Ley de Boyle

Anexo 4. Rúbrica de evaluación.

Valoración	3 puntos Muy bien	2 punto Regular	1 puntos Deficiente
Contenido y profundización de la Investigación	Presenta el tema con profundidad y de manera sintetizada, de forma clara y sustancial.	Presenta el tema pero no profundiza de forma sintetizada o clara y sustancial.	No presenta el contenido completo, su capacidad de síntesis es insuficiente.
Cuadro comparativo señales digitales y analógicas	Las características de comparación son suficientes y pertinentes Se distingue de manera clara y precisa las semejanzas y diferencias entre los elementos comparados	Las características de comparación mínimas e insuficientes Se identifican algunas semejanzas y diferencias entre los elementos comparados	No enuncia las características a comparar. Identifica no de forma clara las semejanzas y diferencias entre los elementos comparados
Desarrollo del experimento	Ilustra con imágenes acordes al contenido de su investigación Ilustra con fotografías el desarrollo de cada etapa del experimento Contesta las preguntas solicitadas realizando justificando sus respuestas	Ilustra con fotografías algunos pasos del desarrollo de cada etapa del experimento Las respuestas no están del todo justificadas	Carece de fotografías o imágenes el trabajo. Sus respuestas carecen de análisis y justificación.
Gráficos	Las gráficas incluyen las tablas de datos respectivos La gráfica coincide bien con los datos y es fácil de interpretar	Las gráficas incluyen sólo algunos datos La gráfica coincide con algunos datos y dificultando su interpretación	No incluye tablas de datos. Los datos graficados no permiten una interpretación de los datos
Conclusiones	Incluye un análisis, da su opinión sustentada.	No elabora una conclusión correcta ni coherente	Sus conclusiones carecen de análisis y sustento.
Organización de la práctica	Trabajo con excelente organización y claramente presentado, así como de fácil seguimiento	Trabajo bien focalizado, pero no suficientemente organizado	Trabajo impreciso y poco claro, sin coherencia entre las partes que lo componen
Sintaxis y ortografía	La redacción es clara y precisa. La ortografía es excelente, no contiene errores de ortografía	La redacción carece de claridad y precisión. La ortografía es buena sin embargo se detectó de 1 a 3 errores de ortografía	La redacción es ambigua. Se detectan más de de 4 faltas de ortografía
Presentación de la práctica	La selección de los colores y la tipografía usada fueron atractivas, además la práctica se entregó de forma limpia	Los colores y la tipografía usada no permiten una correcta visualización	Se abusó del uso de colores y tipografías
Citas y referencias	Registra las referencias en formato APA perfectamente hechas.	Registra las referencias en formato APA no bien hechas (falta algún elemento)	No registra referencias o sólo la dirección URL
Total			