

Mejorando una práctica del laboratorio docente de Física mediante el uso de Arduino y de sensores de bajo coste.

Manuel Domínguez¹, Almudena Aguinaco¹, José A. Ángel¹, Eduardo Blanco¹, Oscar Bomatti¹, Juan M. González-Leal¹, Rocío Litrán¹, José M. Manuel¹, Ignacio Mateos², Manuel Piñero¹, Milagrosa Ramírez¹, Jorge L. Mena-Lorenzo³

¹Departamento de Física de la Materia Condensada, Facultad de Ciencias, ²Departamento de Ingeniería en Automática, Electrónica, Arquitectura y Redes de Computadoras, Escuela Superior de Ingeniería, 11510 Puerto Real (Cádiz, España), ³Departamento de Física, Universidad de Pinar del Río, Pinar del Río (Cuba)

manolo.dominguez@uca.es

RESUMEN: Se ha utilizado una placa Arduino y un conjunto de sensores de bajo coste para estudiar un experimento básico de Física, en el que un carro acelera uniformemente sobre un plano inclinado, usando como base un montaje experimental disponible comercialmente. La precisión de las medidas queda garantizada por el uso de un sensor ultrasónico de distancia acoplado a sensores de temperatura y humedad, con el fin de obtener un valor fiable de la velocidad del sonido durante las medidas. Para completar el sistema de medida de bajo coste, se utilizó una placa Raspberry Pi 3 Modelo B+ a modo de ordenador personal, para controlar la placa Arduino, así como recibir, almacenar y tratar los valores medidos de los parámetros experimentales. Las curvas posición-tiempo, así como los valores de aceleración obtenidos, en función del ángulo de inclinación, están en el rango esperado y demuestran que el efecto de las fuerzas no conservativas (rozamientos, resistencia del aire, etc.) no puede ser descartado. Este sistema de medida experimental de código abierto puede ser fácilmente replicado y aplicado a otros sistemas similares, para favorecer en los estudiantes la comprensión de los conceptos de la Física durante su estancia en el laboratorio de prácticas, así como estimular su interés por la programación científica.

PALABRAS CLAVE (*se indexarán para facilitar la búsqueda de este documento*): proyecto de innovación y mejora docente, prácticas de laboratorio, Física, programación científica, software y hardware libre, Raspberry Pi, Arduino

INTRODUCCIÓN

La importancia de las actividades experimentales para el aprendizaje y la comprensión de los conceptos fundamentales de la Física es un hecho ampliamente reconocido. Cuando un estudiante demuestra una ley física o reconstruye un fenómeno físico en el laboratorio, se facilita su comprensión de éstos de una manera más estable, duradera y funcional que cuando estos conceptos son solamente explicados de forma teórica en clase. Por estas razones, la enseñanza de la Física requiere de laboratorios bien equipados en los que la ciencia y la tecnología puedan mostrar su desarrollo actual y progresivo, algo que resulta extremadamente caro para las instituciones educativas, más aún en países en vías de desarrollo.

En 2005, se anunció en el *Ivrea Interaction Design Institute* la creación de una tarjeta de desarrollo de *hardware* de código abierto denominada Arduino. Se trata de una plataforma electrónica que, aplicando conceptos de la programación con código abierto (*open-source software*), supuso un gran cambio [1, 2], tanto en la calidad como en el desarrollo, adquisición y puesta en marcha de experimentos para los laboratorios docentes, no solo de electrónica, sino también de otras muchas disciplinas científicas (Física, Química, Biología, etc.), gracias al uso de sensores conectados a la propia placa Arduino [3, 4].

Con el uso de Arduino, el tiempo para registrar, procesar y representar gráficamente las magnitudes físicas medidas se puede reducir drásticamente [5], permitiendo obtener resultados experimentales más cercanos a su valor real [6, 7] y aumentando la calidad de la docencia y aprendizaje experimentales. Aún más, el tiempo reducido que se requiere para obtener los datos experimentales puede aprovecharse no solo para que los estudiantes aprendan técnicas básicas de

programación científica (usando un lenguaje de programación similar a C, denominado *Cabling* para Arduino), sino que también permite ahondar en conceptos físicos o expandir el rango y número de fenómenos físicos estudiados en el laboratorio. El menor coste de estos sistemas de medida experimentales, en comparación con las soluciones comerciales disponibles, es claramente beneficioso para las instituciones académicas, pero también una buena oportunidad para que los estudiantes exploren el mundo de las medidas asistidas por ordenador, que se encontrarán más adelante con profusión en su actividad profesional.

En este trabajo describiremos la aplicación de un sistema de medida basado en la placa Arduino para aumentar el número de experimentos físicos que pueden realizarse con un montaje experimental de laboratorio comercial, en concreto un plano inclinado de inclinación variable, sobre el que se sitúa un carro de peso también variable. Originalmente este montaje se utiliza en modo estático, determinándose las fuerzas que actúan sobre el carro mientras se encuentra en equilibrio sobre la plataforma inclinada, gracias a sendos dinamómetros que se fijan en el extremo del plano inclinado y al propio. Con el uso de Arduino, se expandirá el tipo de medida que se realiza con este montaje para estudiar el movimiento del carro sobre el plano inclinado.

Para la descripción cinemática del movimiento de un carro que viaja cuesta abajo sobre un plano inclinado durante un intervalo de tiempo, usamos las magnitudes físicas de posición, velocidad y aceleración. En el esquema del sistema experimental mostrado en la Figura 1, la aceleración del carro tendrá un valor constante si la fricción y otras fuerzas no conservativas son constantes durante el movimiento que el carro realiza sobre el plano inclinado.

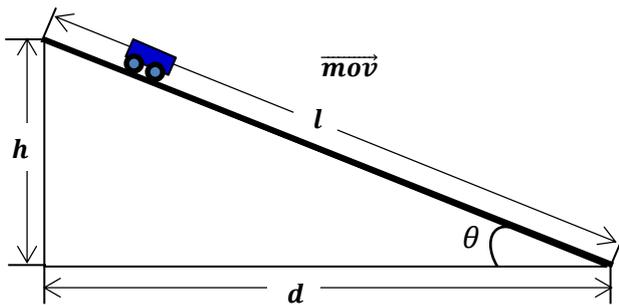


Figura 1. Diagrama cinemático que representa el movimiento cuesta abajo del carro.

En estas condiciones, la velocidad del carro aumenta de forma constante con el tiempo y, por tanto, éste experimenta un movimiento rectilíneo uniformemente acelerado (MRUA). Como el movimiento es unidireccional, una sola coordenada es suficiente para describir la posición como:

$$x = x_0 + \frac{a(t - t_0)^2}{2} \quad (1)$$

En el caso ideal, despreciando las fuerzas no conservativas, el valor estimado de la aceleración del carro, derivado de la aplicación de la segunda ley de Newton, sería

$$a_{est} = g \sin \theta \quad (2)$$

donde θ es el ángulo de inclinación del plano y g es la aceleración de la gravedad en el lugar del experimento. Sin embargo, en el mundo real, la fricción y otras fuerzas no conservativas (la resistencia del aire, por ejemplo) reducen esta aceleración. Como es difícil estimar el efecto y el valor de las fuerzas no conservativas, su efecto combinado se estima a partir de la diferencia entre el valor experimental de la aceleración (a_{exp}) y el caso ideal (a_{est}), obteniendo la expresión:

$$\frac{\sum F_{nc}}{m_c} = a_{est} - a_{exp} \quad (3)$$

MONTAJE EXPERIMENTAL DE MEDIDA

El sistema experimental mostrado en la Figura 2 se basa en una práctica de laboratorio comercializada en España por la compañía *Ventus*, que incluye un carro de masa variable con ruedas de baja fricción, además del plano inclinado de inclinación variable mostrados. El objetivo original de este montaje es medir, con un dinamómetro, las fuerzas que actúan sobre el carro, dependiendo de su masa y de la inclinación de la rampa. Es, en esencia un experimento de estática que, mediante el uso de Arduino y un conjunto de diferentes sensores de bajo coste, se ha transformado en un experimento de cinemática. El plano inclinado está hecho de aluminio, con una longitud de 60 cm, y posee una bisagra y un sistema de fijación para establecer el ángulo de inclinación, que puede medirse con ayuda de un transportador de ángulos que posee una precisión de 1°. El carro de 180 g, tiene cuatro ruedas que incorporan cojinetes de baja fricción, y puede cargarse con pesas adicionales (hasta cuatro de 50 g). En el extremo superior del plano inclinado, una polea de baja fricción facilita la conexión del carro a un peso colgante de masa variable. Este sistema es utilizado en otra versión de esta práctica de laboratorio que explora el movimiento ascendente del carro.

En este sistema experimental expandido con Arduino, la adquisición de datos de posición del carro en función de tiempo

se realiza con ayuda de un sensor ultrasónico HC-SR04 que se conecta a una tarjeta Arduino UNO. Además, la placa Arduino se conecta a un ordenador Raspberry Pi 3 Modelo B+, que actúa como ordenador personal, para conformar el sistema de medida de bajo coste.

De acuerdo con su hoja de características, este sensor de ecolocalización es capaz de medir distancias de objetos situados entre 2 y 400 cm, con una precisión de 3 mm y un cono de dispersión angular del haz de 15°. Trabaja a 15 mA y 40 kHz con un voltaje de operación de 5 V. Sus cuatro pines se conectan de la siguiente manera: el pin *Vcc* al puerto de 5V del Arduino, el pin *Gnd* a la conexión de tierra del Arduino (*ground*), el pin *Trig* (disparo del impulso ultrasónico) al pin 8 del Arduino y el pin *Echo* (recepción del pulso de eco) al pin 12 del Arduino. Este sensor ultrasónico trabaja enviando una señal de disparo de 5 V de al menos 10 μ s para emitir una ráfaga de ultrasonidos de 8 ciclos a 40 kHz que, tras reflejarse en el objeto que se debe localizar, se recibe en el sensor como una señal de eco. El emisor y el receptor ultrasónico están separados 2.6 cm en la placa del sensor. La distancia entre el sensor y el objeto se calcula a partir del tiempo transcurrido entre la emisión de la señal y la recepción del eco, considerando la velocidad del sonido. Sin embargo, aunque no se tiene en cuenta en la mayoría de las aplicaciones de este tipo de sensores, la velocidad del sonido depende significativamente de varios factores ambientales, principalmente la temperatura y la humedad.

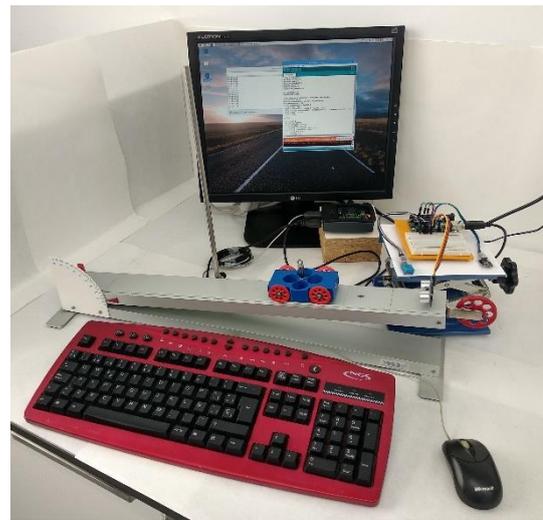


Figura 2. Sistema experimental compuesto por el plano inclinado de origen comercial, junto con las placas Arduino y Raspberry Pi.

Por esta razón, hemos añadido a nuestro Sistema experimental un sensor de temperatura analógico (KY-013) y un sensor de humedad relativa (KY-015, basado en el sensor digital combinado de temperatura y humedad DHT11). El módulo sensor de temperatura KY-013 consiste en un termistor NTC con una resistencia de 10 k Ω . El valor de la resistencia del termistor varía con la temperatura ambiente, así que se usó la ecuación simplificada de Steinhart-Hart [8] para deducir los valores precisos de temperatura derivados de la señal del termistor

$$\frac{1}{T} = A + B \ln R + C (\ln R)^3 \quad (4)$$

donde T es la temperatura en Kelvin, R es la Resistencia eléctrica del termistor en ohmios, y A , B y C , son los

denominados coeficientes de Steinhart-Hart. El rango de temperatura en el que opera este sensor KY-013 está entre -55 y 125°C con una precisión de 0.5°C . Su voltaje de operación es de 5V , que también proporciona la placa Arduino, y sus otros dos pines se conectan a tierra y al pin A0 del Arduino. En cualquier caso, es conveniente calibrar este tipo de sensores de bajo coste, puesto que los valores nominales de los parámetros de Steinhart-Hart encontrados en sus hojas de características pueden estar alejados de los correctos. Por ello, el sensor se embebó en un tubo termoretráctil y así se calibró, midiendo la resistencia del termistor en tres valores de temperatura: la ambiente, la de un baño de hielo fundente (aproximadamente, 0°C) y la de un baño de agua destilada en ebullición (aproximadamente, 100°C). Se utilizó para ello un sensor independiente calibrado de temperatura. Los valores de la resistencia del termistor en los tres casos se usaron para obtener los coeficientes de Steinhart-Hart del sensor utilizando una calculadora en línea [9].

Como se ha indicado con anterioridad, el módulo KY-015 es un sensor digital combinado de temperatura y humedad relativa. En este caso, se excluyó el sensor de temperatura en favor del sensor KY-013 por su menor precisión ($\pm 2^{\circ}\text{C}$). Aunque la precisión del sensor de humedad relativa es de solo $\pm 5\%$, esta cifra es suficiente para nuestra aplicación. Los tres pines del sensor KY-015 se conectaron de la siguiente forma: el pin (+) al pin 5V del Arduino, el pin (-) al pin tierra del Arduino, y la salida digital al pin 2 del Arduino. El diagrama completo de conexiones con la placa Arduino UNO y una placa de desarrollo usada aquí se muestra en la Figura 3 [10].

En la parte central del plano inclinado existen varios agujeros que se usan en la práctica original para fijar, por medio de unos clips de plástico, sujetos a una conector banana, los dinamómetros que sirven para medir las fuerzas que actúan sobre el carro en reposo. En nuestro caso, el agujero de la parte superior se utilizó para insertar un conector banana pegado al sensor ultrasónico HC-SR04.

El código Arduino (desarrollado en el IDE de *Cabling*) para esta medida se muestra en el Anexo I y se usó para que, desde una interfaz amigable desarrollada en *Python*, que corría sobre la Raspberry Pi, se determinaran y guardaran en un archivo de texto los datos experimentales de las medidas de la posición del carro (valores x) en función del tiempo (valores t). Se usaron las librerías *NewPing* [11] y *SimpleDHT* [12], respectivamente, para obtener de forma continua los datos del intervalo entre la señal ultrasónica emitida y su eco, así como el valor de la humedad relativa. Con posterioridad, los datos almacenados en la memoria de la Raspberry Pi se representaron gráficamente y se analizaron con el propio software instalado en la Raspberry Pi (una hoja de cálculo de código abierto). Como puede deducirse del código del Anexo I, la distancia x se calculó a partir del intervalo de tiempo entre la señal emitida y el eco, aplicando al cálculo la velocidad del sonido (v , en m/s) que se calcula de la bien conocida expresión:

$$v = 331.3 + 0.6 \cdot T + 0.0124 \cdot H \quad (5)$$

siendo T la temperatura en $^{\circ}\text{C}$ y H , la humedad relativa en tanto por ciento, medidas con los módulos sensores KY-013 y KY-015, respectivamente. El valor de la distancia se corrige para tener en cuenta la separación entre el emisor y el receptor en el sensor ultrasónico HC-SR04, usando el teorema de Pitágoras [13].

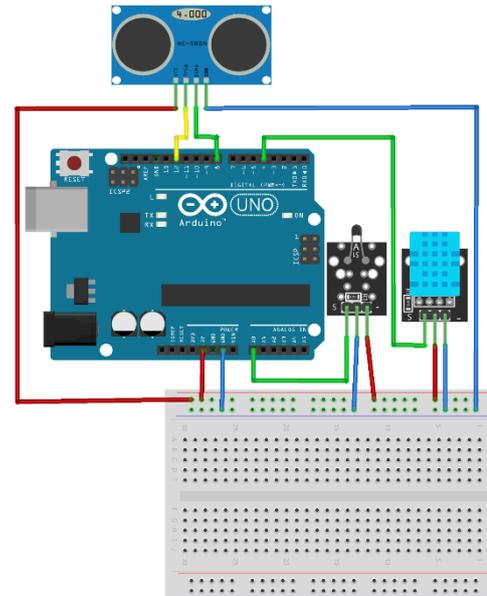


Figura 3. Esquema de conexiones entre la placa Arduino UNO y los sensores de posición, temperatura y humedad.

RESULTADOS EXPERIMENTALES

A modo de ejemplo, se presentarán a continuación los datos experimentales para tres inclinaciones del plano (2 , 4 y 6°), mostrando los resultados en la Figura 4. Las líneas discontinuas corresponden a los ajustes no lineales a la ecuación (1) en cada caso. Dado que la posición inicial x_0 se elige arbitrariamente y que el instante inicial t_0 también se establece de forma aleatoria por el usuario, ambos valores son desconocidos inicialmente. Por ello, los ajustes de los valores experimentales posición-tiempo a la ecuación (1) deben realizarse con un algoritmo de ajuste iterativo. En este caso empleamos una herramienta que tienen a su disposición de los estudiantes: la herramienta *Solver* que puede encontrarse tanto en hojas de cálculo libres como LibreOffice Calc, como en sus equivalentes de software propietario. Por medio de este ajuste no lineal, se obtienen simultáneamente los valores de x_0 , t_0 and a_{exp} . Al mismo tiempo, esta es una buena oportunidad para introducir a los estudiantes el concepto de ajuste no lineal, así como las herramientas computacionales a su alcance para llevarlo a cabo.

Los valores de la aceleración del carro se muestran en la Tabla 1 para los tres ángulos de inclinación citados, incluyendo también la comparación con el valor de la componente tangencial de la aceleración de la gravedad, que sería el valor esperado en condiciones ideales, sin fuerzas no conservativas. Se deduce así la diferencia entre ambas, como indica la ecuación (3).

Puesto que las fuerzas no conservativas pueden verse afectadas por varios factores aparte del ángulo de inclinación, tales como el área de contacto entre las ruedas y la rampa o la dependencia exacta de la fuerza de resistencia del aire o fuerza de arrastre con la velocidad del carro, no es sencillo encontrar una dependencia directa de a_{nc} del ángulo de inclinación. En cualquier caso, la discusión sobre el origen e influencia de las fuerzas no conservativas sobre la diferencia observada entre el valor ideal y el experimental de la aceleración, se sugeriría a los alumnos para que lo incluyan en sus informes de laboratorio.

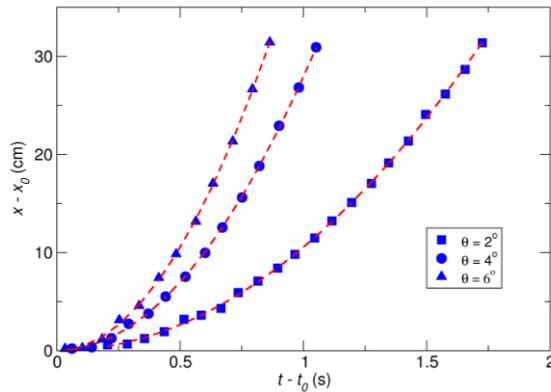


Figura 4. Resultados experimentales obtenidos con 3 ángulos de inclinación. Las líneas discontinuas corresponden a los ajustes a la ecuación (1).

Table 1. Comparación entre los valores de aceleración obtenidos de los ajustes a la ecuación (1) en la Figura 4 y los valores ideales estimados. La diferencia se debe a las fuerzas no conservativas.

θ (°)	a_{exp} (cm/s ²)	a_{est} (cm/s ²)	$\Sigma F_{nc} / m_c$ (cm/s ²)
2	21.1	34.2	13.1
4	55.8	68.4	12.6
6	84.4	102.4	18.0

CONCLUSIONES

La adición de hardware de Código abierto, basado en el uso de una placa Arduino junto con sensores de bajo coste, posibilita la expansión del número y del tipo de experimentos que se pueden realizar con un sistema experimental comercializado para las prácticas de laboratorio de Física. Estos sensores de bajo coste, tras una adecuada calibración, permiten obtener medidas razonablemente precisas, a partir de las que los estudiantes pueden obtener una visión más amplia para ayudarles a comprender los detalles más significativos de las leyes de la Física en el laboratorio. Además, al exponer a los estudiantes al uso de sistemas de medida basados en Arduino se puede inducir en ellos la propuesta de nuevos experimentos usando sensores similares, así como la posibilidad de adquirir conocimientos de programación científica, que pueden serles de utilidad más adelante.

El sistema experimental descrito aquí, que utiliza un sensor ultrasónico junto con sensores de temperatura y humedad, empleado para estudiar la cinemática del movimiento en un plano inclinado, es sólo un ejemplo de las posibilidades que estos sistemas posibilitan para la mejora y transformación de las prácticas de laboratorio de Física como herramientas aún más útiles de cara la formación de nuestros estudiantes.

REFERENCIAS

1. Kushner, D. The Making of Arduino. *IEEE Spectrum*. 2011, <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>. Último acceso el 25 de septiembre de 2019.
2. Kinchin, J. Using an Arduino in physics teaching for beginners. *Phys. Educ.* **2018**, *53*, 063007.
3. Banzi, M. *Getting Started with Arduino*. O'Reilly (USA) **2011**.

4. Moya, A. A. An Arduino experiment to study free fall at schools. *Phys. Educ.* **2018**, *53*, 055020.

5. Liu C., Wu C., Wong W., Lien Y. y Chao T. Scientific modelling with mobile devices in high school physics labs. *Comput. Educ.* **2017**, *105*, 44–56.

6. Sari, U. Using the Arduino for the experimental determination of a friction coefficient. *Phys. Educ.* **2019**, *54*, 035010.

7. Pili, U. y Violanda R. Measurement of the gravitational acceleration using a simple pendulum apparatus, ultrasonic sensor, and Arduino. *Phys. Educ.* **2019**, *54*, 043009.

8. Steinhart, J.S. y Hart, S.R. Calibration curves for thermistors. *Deep-Sea Research.* **1968**, *15*, 497-503

9. Los coeficientes Steinhart-Hart se obtuvieron mediante la calculadora en línea disponible en el sitio web: <https://www.thinksrs.com/downloads/programs/therm%20calc/ntccalibrator/ntccalculator.html>. Último acceso el 15 de julio de 2019.

10. La imagen realizada con el programa de código abierto *Fritzing* que puede descargarse de: <http://fritzing.org/home/>. Último acceso el 15 de julio de 2019.

11. La librería NewPing se puede descargar del sitio web <https://playground.arduino.cc/Code/NewPing/>. Último acceso el 20 de julio de 2019.

12. La librería SimpleDHT puede descargarse del sitio web <https://arduinolibraries.info/libraries/simple-dht>. Último acceso el 20 de julio de 2019.

13. Goncalves, A.M.B. Cena, C.R. y Bozano, D.F. Driven damped harmonic oscillator resonance with Arduino. *Phys. Educ.* **2017**, *52*, 055020.

ANEXOS

sol-201800111587-tra_Anexo 1.pdf

AGRADECIMIENTOS

Todos los autores agradecen la financiación del Vicerrectorado de Recursos Docentes y de la Comunicación, proyecto sol-201800111587-tra. J.L.M.L. agradece la financiación del Aula Universitaria Iberoamericana para su estancia en la Universidad de Cádiz.

```

//Kinematics in the Inclined Plane
#include <NewPing.h> //Load HC-SR04 sensor library
#include <math.h> //Load Mathematics library
#include <SimpleDHT.h> //Load DHT11 sensor library
#define trigPin 8 //Declare ultrasound triggering pin 8
#define echoPin 12 //Declare input ultrasound echo pin 12
#define max_distance 400 //Declare maximum distance measured with HC-SR04
int pinDHT11 = 4; //Declare humidity sensor input pin 4
SimpleDHT11 dht11; //Declare humidity sensor type

int ThermistorPin = A0; //Declare input pin for the analog temperature sensor A0
double Thermistor(int RawADC) { //Declare function Thermistor
  double LogR, Temp, temp; //Declare temperature related variables as double
  LogR = log(10000.0 * ((1024.0 / RawADC - 1))); //Calculate log(R) from KY-013 raw output
  Temp = 1 / (0.0009723324827 + (0.0002626077614 - (0.00000003766897239 * LogR * LogR)) * LogR); //Calculate the temperature in K with theSteinhart-Hart equation
  temp = Temp - 273.15; //Convert T from K to °C
  return temp; //Return T in °C
}

NewPing sonar(trigPin, echoPin, max_distance); //Initialize NewPing to use trigger and echo pins with maximum distance (optional)

unsigned long exp_time = 0; //Declare time of experiment variable as unsigned long setting it to 0
unsigned long current_exp_time = 0; //Declare current time of experiment as unsigned long setting it to 0
const long interval = 100; //Declare interval time between measurements as const long setting it to 100 ms
int hr; //Declare variable for relative humidity as integer

float distance_d, distance_x, echo_time, speedSound; //Declare variables as float

void setup() {
  Serial.begin(9600); //Initialize serial monitor
  pinMode(trigPin, OUTPUT); //Declare trigPin as digital output
  pinMode(echoPin, INPUT); //Declare echoPin as digital input
  int readVal = analogRead(ThermistorPin); //Read KY-013 output
  double temp = Thermistor(readVal); //Calculate the room temperature from KY-013 by using function Thermistor
  byte temperature = 0; //Declare KY-015 digital temperature variable as byte
  byte humidity = 0; //Declare KY-015 digital humidity variable as byte
  byte data[40] = {0}; //Declare data array as byte
  if (dht11.read(pinDHT11, &temperature, &humidity, data)) {
    Serial.println("DHT11 Reading error"); //Check for DHT reading errors
    return;
  }
  hr = ((int)humidity); //Set variable hr to the relative humidity value read from sensor KY-015

  speedSound = 331.3 + 0.606 * temp + 0.0124 * hr; //Calculate the speed of sound using temperature and humidity values
}
//Position-time measurements loop
void loop() {
  exp_time = millis(); //Measurement time in milliseconds
  if ( exp_time > current_exp_time + interval) { //Perform measurement if interval between measurements have passed
    current_exp_time = exp_time; //Set current experiment time to experiment time
    echo_time = sonar.ping(); //Obtain the time elapsed between ping and echo to calculate object position
    exp_time = millis(); //Correct measurement time in milliseconds
    distance_d = echo_time / 2.0 * (speedSound / 10000); //Calculate diistance between the ultrasonic sensor and the object
    distance_x = sqrt(pow(distance_d, 2) - pow(1.3, 2)); //Correction due distance between the emitter and receptor on HC-SR04
    Serial.print(exp_time / 1000000.0); //Print the measurement time in seconds
    Serial.print(" "); //Print space as separator
    Serial.println(distance_x); //Print the position of the object in cm
  }
}
}

```

Figura A1. Código Arduino desarrollado para el control y medida del sistema experimental.