**INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO**





Laboratorio de Principios de Mecatrónica

**Práctica 3. Sensores**

Estudiantes:

* Plauchú Rodríguez Rodrigo 182671
* Castillejos Corzo Victor Hugo 182344

Asignatura: Laboratorio de Principios de Mecatrónica

Docente: M.I. Sergio Hernández Sánchez

Grupo: \_04\_

1. **Introducción**

Esta práctica consiste en desarrollar las habilidades necesarias y crear conocimiento suficiente para comprender el funcionamiento de los sensores de movimiento, calor, luminosidad, humedad, etc. A continuación, se analizará el comportamiento de diversos sensores y cómo utilizarlos.

1. **Objetivos**

* Desplegar en texto y de manera gráfica la lectura obtenida del sensor.
* Obtener la señal de voltaje analógico que emite cada sensor por medio del canal ADC y correlacionarla.
* Instalar e incluir bibliotecas en el IDE de Arduino
* Proponer aplicaciones de los sensores estudiados

1. **Marco teórico**

Joystick analógico arduino: Está construido mediante un conjunto de resistencias variables. Es una herramienta muy útil para darle dirección a las cosas (videojuegos, autos, etc.). El funcionamiento está basado en el plano cartesiano y el movimiento es capturado por 2 potenciómetros qué cada uno representa un eje. Además, tiene 5 pines diferentes (GND, +5V, pinX, pinY, pinPush). Los primeros dos son para alimentar el circuito y los 3 restantes son para recibir las señales dadas.

Divisor de voltaje: Es un conjunto de resistencias conectadas en serie. Se toma el voltaje de una resistencia y no de las 2. Por las leyes de Ohm, sabemos qué v=i\*R y el voltaje en serie es: Vin=i\*(R1+R2). Por lo tanto, Vout=Vin \*R2/(R1+R2). Por lo anterior, podemos concluir que el voltaje se divide.

LDR: Es un fotoresistor que su resistencia varía dependiendo de la luz qué recibe. “Un fotoresistor está formado por un semiconductor, típicamente sulfuro de cadmio CdS. Al incidir la luz sobre él algunos de los fotones son absorbidos, provocando que electrones pasen a la banda de conducción y, por tanto, disminuyendo la resistencia del componente.

Por tanto, un fotoresistor disminuye su resistencia a medida que aumenta la luz sobre él. Los valores típicos son de 1 Mohm en total oscuridad, a 50-100 Ohm bajo luz brillante.” (Luis, 2016)

DHT11: Es un sensor de temperatura y humedad que es digital. Esto nos sirve mucho porque nos facilita la lectura de las mediciones que hacemos. Por lo tanto, hay menos ruido en las lecturas. Por lo anterior, es un sensor muy fiable y estable. Tiene una resistencia pull-up de 5kOhms. Sus pines son alimentación, tierra y la transmisión de datos.

1. **Material y equipo utilizado**

* 1 Arduino MEGA
* 1 cable USB A/B
* 1 Potenciómetro de 10 kΩ
* 1 *Display* LCD 16X2
* 1 Resistor 220 Ω
* 1 Joystick Analógico
* 1 resistor LDR
* 1 Sensor de temperatura y humedad DTH11

1. **Experimentos**
2. **Actividad 1 – Joystick analógico**

|  | Defina según la imagen, si este es el eje X o Y  **Eje: X** |  | |
| --- | --- | --- | --- |
| Coloque el valor que obtiene si coloca la perilla completamente a hacia arriba:  **Valor: 1023** |
| Coloque el valor que obtiene si coloca la perilla completamente a la izquierda:  **Valor: -512** |  | Coloque el valor que obtiene si coloca la perilla completamente a la derecha:  **Valor: 512** |  |
|  | Valores cuando el joystick está en reposo  **X: 0 Y: 0** |  |  |
| Coloque el valor que obtiene si coloca la perilla completamente a hacia abajo:  **Valor: -1023** |  | |

**//ACT 1.2**

**//Se definen puertos**

**#define X A0**

**#define Y A1**

**#define BTN 13**

**//declaramos variables**

**float xPos, yPos;**

**bool button;**

**void setup() {**

**// put your setup code here, to run once:**

**//communications at 9600 bps:**

**Serial.begin(9600);**

**// Initialize pin modes:**

**pinMode(BTN, INPUT\_PULLUP);**

**}**

**void loop() {**

**// put your main code here, to run repeatedly:**

**// Se leen los valores**

**xPos = analogRead(X);**

**yPos = analogRead(Y);**

**button = digitalRead(BTN);**

**// Imprime (x, y) values:**

**Serial.print("Valores (x, y) registrados por el joystick entre -1, 1: ");**

**//Serial.print("(");**

**Serial.println((xPos-512)/512);**

**//Serial.print(", ");**

**Serial.println((512-yPos)/512);**

**//Serial.println(" ");**

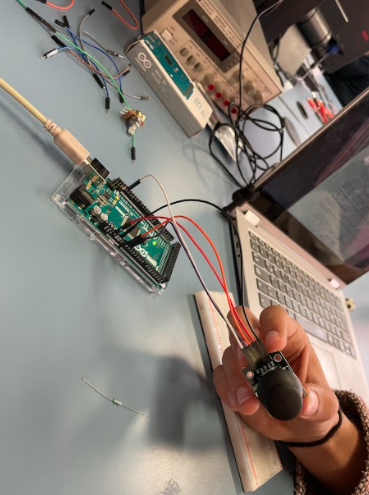
**Serial.println(button);**

**//Serial.println(")");**

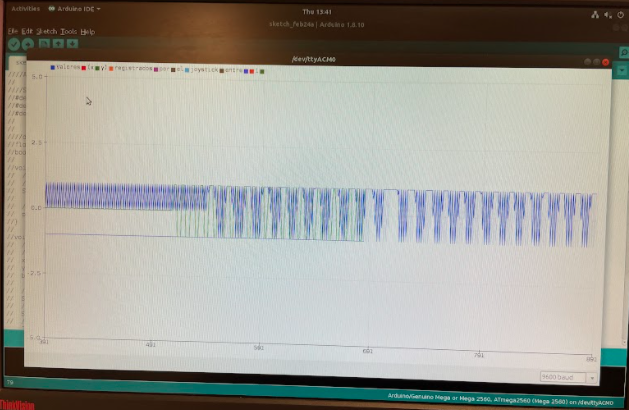
**delay(100);**

**}**

**Armado**

****

**Serial Plotter**

****

1. **Actividad 2 – Sensor de luminosidad**

**ACT 3**

**#define X A3**

**#define LED 13**

**float x;**

**void setup () {**

**Serial.begin(9600);**

**// Initialize pin modes:**

**pinMode(LED, OUTPUT);**

**}**

**void loop () {**

**x = analogRead(X);**

**Serial.println(x);**

**if (x < 90) {**

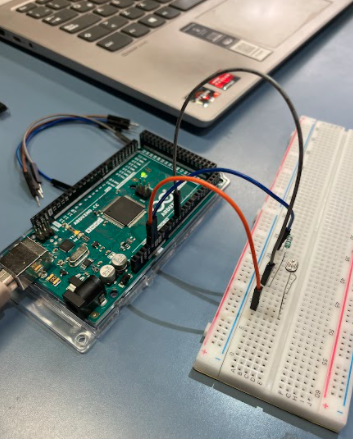
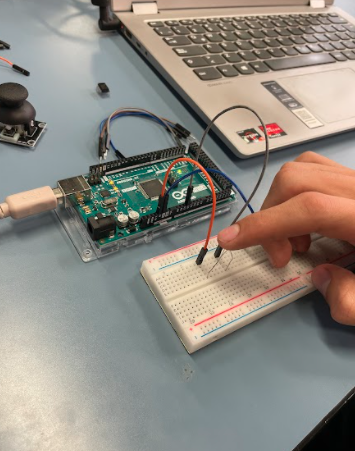
**digitalWrite(LED, HIGH);**

**} else {**

**digitalWrite(LED, LOW);**

**}**

**}**

****

**Led apagado y prendido**

1. **Actividad 3 – Sensor de temperatura y humedad**

**//ACT TEMP Y HUMEDAD**

**#include <DHT.h>**

**#include <LiquidCrystal.h> // inluimos la lib**

**DHT dht(7, DHT11);**

**LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // initialize the interface pins**

**void setup() {**

**// put your setup code here, to run once:**

**lcd.begin(16, 2);**

**Serial.begin(9600);**

**dht.begin();**

**}**

**void loop() {**

**// put your main code here, to run repeatedly:**

**float hum = dht.readHumidity();**

**float tempC = dht.readTemperature();**

**float tempF = dht.readTemperature(true);**

**Serial.print("Humedad: ");**

**Serial.print(hum);**

**Serial.print(" Temp: ");**

**Serial.print(tempC);**

**Serial.print("ºC \_ ");**

**Serial.print(tempF);**

**Serial.println("ºF");**

**//lcd**

**lcd.clear();**

**lcd.setCursor(0, 0);**

**lcd.print("Temp C: Hum:"); // Print a message to the LCD.**

**lcd.setCursor(0, 1);**

**lcd.print(tempC); // Print a message to the LCD.**

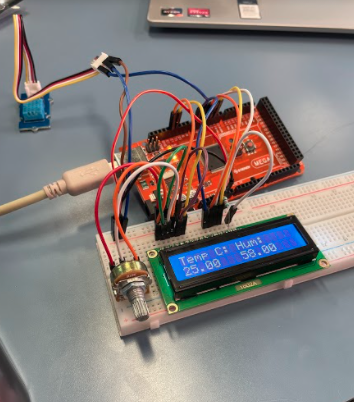
**lcd.setCursor(8, 1);**

**lcd.print(hum); // Print a message to the LCD.**

**// Esperamos un tiempo para repetir el loop**

**delay(1000);**

**}**

****

1. **Actividad 4 – Medición de inclinación**

*Tabla 1. Obtención de los valores máximos y mínimos del acelerómetro.*

|  | **Eje X** | **Eje Y** | **Eje Z** |
| --- | --- | --- | --- |
| **Mínimo** | 260 | 270 | 297 |
| **Máximo** | 410 | 400 | 435 |

**//#include <LiquidCrystal.h> // include the library code**

**// Declaracion de variables globales**

**//LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // initialize the interface pins**

**//int t\_muestreo = 500; // Tiempo de muestreo del acelerómetro**

**int xpin = A0;**

**int ypin = A1;**

**int zpin = A2;**

**int minVal = 0; // Valor mínimo del acelerometro en reposo**

**int maxVal = 1023; // Valor máximo del acelerometro en reposo**

**void setup() {**

**// put your setup code here, to run once:**

**Serial.begin(9600); // Configuramos el puerto serial a 9600 bps**

**//lcd.begin(16, 2); // set up the LCD’s number of columns and rows**

**//analogReference(EXTERNAL); // La tensión aplicada en el pin AREF (3.3V) será la que haga que el conversor analogo-digital de su máxima lectura (1023)**

**// Initialize pin modes:**

**}**

**void loop() {**

**// put your main code here, to run repeatedly:**

**// Mostrar en el LCD**

**// lcd.clear();**

**// lcd.setCursor(2, 0);**

**// lcd.print("X Y Z"); // Print a message to the LCD.**

**int xR = analogRead(xpin); // Leemos el valor de la tensión en el pin x**

**int yR = analogRead(ypin); // Leemos el valor de la tensión en el pin y**

**int zR = analogRead(zpin); // Leemos el valor de la tensión en el pin z**

**// Mapea los valores leidos a un rango -90 a 90 grados (-pi a pi)**

**int x\_ang = map(xR, minVal, maxVal, -90, 90);**

**int y\_ang = map(yR, minVal, maxVal, -90, 90);**

**int z\_ang = map(zR, minVal, maxVal, -180, 180);**

**Serial.print("X: ");**

**Serial.print(xR);**

**Serial.print(" Y: ");**

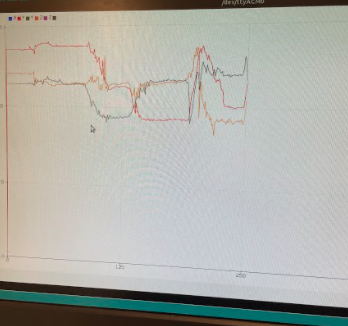
**Serial.print(yR);**

**Serial.print(" Z: ");**

**Serial.println(zR);**

**delay(100);**

**}**

****

**NOTA: Se indicó que la segunda parte de esta actividad, así como la actividad 5 serán terminadas y entregadas en la siguiente sesión.**

1. **Conclusiones**

Sabemos que aún no culmina totalmente esta práctica, sin embargo, por el momento hemos cumplido los objetivos teóricos y prácticos. Particularmente, hemos obtenido buenos resultados y hemos empezado a notar una familiarización con los componentes y con arduino, lo cual nos hace cometer cada vez menos errores. Conocimos componentes muy interesantes que no sabíamos cuál era su composición interna como el sensor de luminosidad o humedad. Finalmente, sobrepasamos los objetivos de esta práctica y esperamos poderla terminar sin nimgún inconveniente y seguir aprendiendo de principios de mecatrónica.

1. **Referencias**

Alexis. (2016, enero 25). Joystick analógico programado con Arduino. - HETPRO/TUTORIALES. *HETPRO/TUTORIALES*. <https://hetpro-store.com/TUTORIALES/joystick-analogico-programado-con-arduino/>

del Valle Hernández, L. (2017, marzo 21). *Cómo utilizar el DHT11 para medir la temperatura y humedad con Arduino*. Programar fácil con Arduino.

<https://programarfacil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>

*Divisor de voltaje*. (s/f). Khan Academy. Recuperado el 9 de marzo de 2022, de

<https://es.khanacademy.org/science/electrical-engineering/ee-circuit-analysis-topic/ee-resistor-circuits/a/ee-voltage-divider>

Luis. (2015, marzo 16). *Medir nivel de luz con Arduino y fotoresistencia LDR (GL55)*. Luis Llamas.

<https://www.luisllamas.es/medir-nivel-luz-con-arduino-y-fotoresistencia-ldr/>