**INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO**





Laboratorio de Principios de Mecatrónica

**Práctica 4. Actuadores**

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**

En esta práctica observaremos cómo se utilizan los actuadores. Para ser específicos, se utilizará un motor de corriente directa con escobillas. Se observará el control de giro, la variación de velocidad de este, y los estados qué puede tener el motor.

1. **Objetivos**

* Conocer los distintos tipos de actuadores que se utilizan en los sistemas mecatrónicos, especialmente los electromecánicos.
* Realizar las conexiones necesarias para operar un motor eléctrico de corriente directa con escobillas en función de sus especificaciones.
* Analizar las señales necesarias para controlar el sentido de giro y la velocidad de dicho motor.
* Aplicar un control proporcional a la velocidad del motor de CD a partir de una señal de entrada mediante una placa de desarrollo Arduino.

1. **Marco Teórico**

* Motor a pasos: “...es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de girar una cantidad de grados (paso o medio paso) dependiendo de sus entradas de control.” (Motor paso a paso, 2021) Gira utilizando imanes permanentes.
* Motor de corriente directa con escobillas: es un motor qué consiste en cuatro partes principales (estator, rotor, conmutador, escobillas). Las escobillas son las encargadas de conmutar la corriente de las bobinas del motor, es decir, cambiar la polaridad del rotor. Este motor es muy económico. Sin embargo, cómo hay fricción, pierde potencia, a diferencia del motor brushless (cfr. Diferencias entre motores con escobillas y brushless, 2016).
* Motor de corriente directa sin escobillas: a diferencia del motor anterior, este no tiene escobillas. Por lo tanto, la conmutación de las bobinas se realiza electrónicamente por un controlador de motor. La vida útil y la velocidad de este motor es mayor al anterior.
* Puente H: “... es un [circuito electrónico](https://es.wikipedia.org/wiki/Circuito_electr%C3%B3nico) que generalmente se usa para permitir a un [motor eléctrico DC](https://es.wikipedia.org/wiki/Motor_de_corriente_continua) girar en ambos sentidos, *avance* y *retroceso*. Son ampliamente usados en [robótica](https://es.wikipedia.org/wiki/Rob%C3%B3tica) y como convertidores de potencia. Los puentes H están disponibles como [circuitos integrados](https://es.wikipedia.org/wiki/Circuito_integrado), pero también pueden construirse a partir de componentes discretos.”(Colaboradores de los proyectos Wikimedia, 2008)
* L293D: es un circuito integrado para controlar motores. Está construido con partes de un puente H. Nos permite controlar distintas cantidades de motores. Puede controlar 4 motores de DC unidireccionalmente. Por otro lado, puede controlar 2 motores en ambas direcciones . Permite una alimentación independiente para los motores. (cfr. Dice, 2017)

1. **Material y equipo utilizado**

* 1 Arduino MEGA
* 1 cable USB A/B
* 1 fuente de voltaje regulada
* 1 par de cables banana caimán
* 1 protoboard
* 1 motor de corriente directa con escobillas y reductor
* 1 puente H L293D
* 2 resistores 220 Ω
* 2 LED’s
* 1 potenciómetro

1. **Experimentos**
   1. **Actividad 1 – Conexión y prueba de un puente H**

**//ACT1**

**#define MotA 10 //se definen pines**

**#define MotB 11**

**void setup() {**

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

**pinMode(MotA, OUTPUT); // pin del led rojo1 como salida**

**pinMode(MotB, OUTPUT); // pin del led rojo2 como salida**

**Serial.begin(9600);**

**}**

**void loop() {**

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

**//Mandamos 0 y 0**

**digitalWrite(MotA, LOW);**

**digitalWrite(MotB, LOW);**

**Serial.print("Freno pasivo: ");**

**Serial.print(MotA);**

**Serial.println(MotB);**

**delay(1000);**

**//Mandamos 0 y 1**

**digitalWrite(MotA, LOW);**

**digitalWrite(MotB, HIGH);**

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

**Serial.print(MotA);**

**Serial.println(MotB);**

**delay(1000);**

**//Mandamos 1 y 0**

**digitalWrite(MotA, HIGH);**

**digitalWrite(MotB, LOW);**

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

**Serial.print(MotA);**

**Serial.println(MotB);**

**delay(1000);**

**//Mandamos 1 y 1**

**digitalWrite(MotA, HIGH);**

**digitalWrite(MotB, HIGH);**

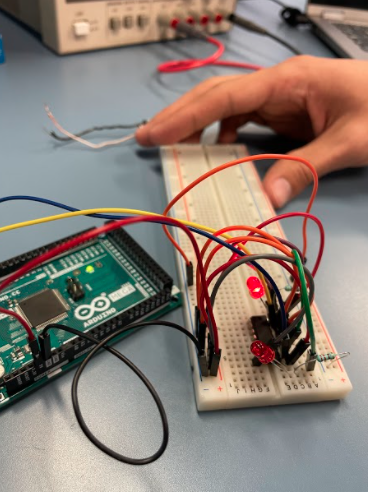
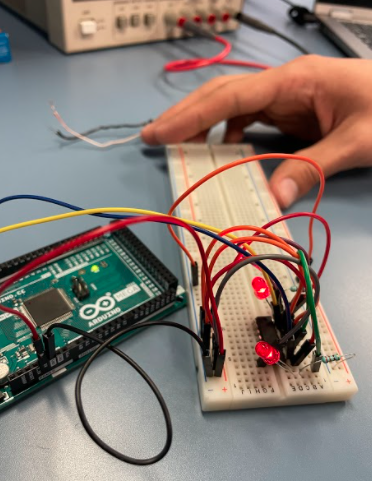
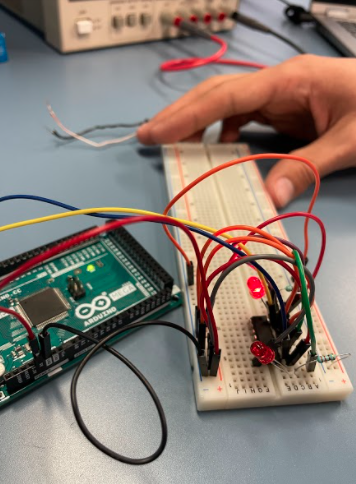
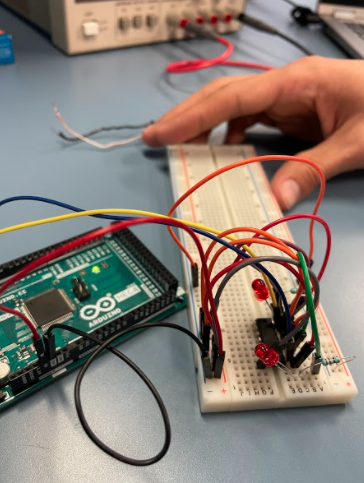
**Serial.print("Freno activo: ");**

**Serial.print(MotA);**

**Serial.println(MotB);**

**delay(1000);**

**}**

****

* 1. **Actividad 2 – Control de giro**

*Tabla 1. Tabla de estados.*

| **Estado** | **Voltaje del potenciómetro** | **Acción del motor** |
| --- | --- | --- |
| 1 | 0V < V\_leído < 2V | Giro levógiro |
| 2 | 2V < V\_leído < 3V | Giro detenido |
| 3 | 3V < V\_leído < 5V | Giro dextrógiro |

**//ACT2**

**#define MotA 10 //se definen pines**

**#define MotB 11**

**#define A0 A0**

**void setup() {**

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

**pinMode(MotA, OUTPUT); // pin del led rojo1 como salida**

**pinMode(MotB, OUTPUT); // pin del led rojo2 como salida**

**Serial.begin(9600);**

**}**

**void loop() {**

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

**// se lee el voltaje del potenciómetro y se convierte a voltaje**

**float x = analogRead(A0)/204.6;**

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

**Serial.println(x);**

**// Se hace el giro cómo se muestra en la tabla. Usamos freno pasivo.**

**if(x >= 0 && x < 2){**

**// Gira a la izquierda**

**levogiro();**

**} if (x >= 2 && x < 3){**

**// frena**

**frenoP();**

**} if (x >= 3 && x <= 5){**

**// Gira a la derecha**

**dextrogiro();**

**}**

**}**

**void levogiro(){**

**//Mandamos 1 y 0**

**digitalWrite(MotA, HIGH);**

**digitalWrite(MotB, LOW);**

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

**Serial.print(MotA);**

**Serial.println(MotB);**

**delay(1000);**

**}**

**void frenoP(){**

**//Mandamos 0 y 0**

**digitalWrite(MotA, LOW);**

**digitalWrite(MotB, LOW);**

**Serial.print("Freno pasivo: ");**

**Serial.print(MotA);**

**Serial.println(MotB);**

**delay(1000);**

**}**

**void dextrogiro(){**

**//Mandamos 0 y 1**

**digitalWrite(MotA, LOW);**

**digitalWrite(MotB, HIGH);**

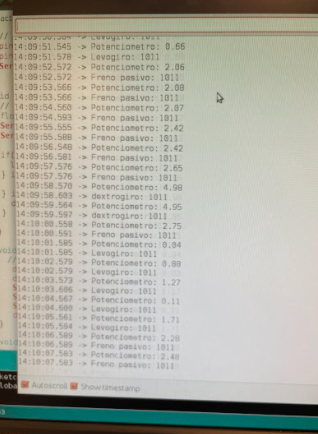
**Serial.print("dextrogiro: ");**

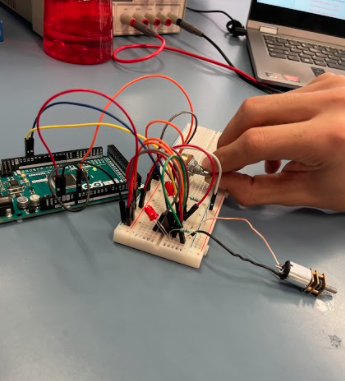
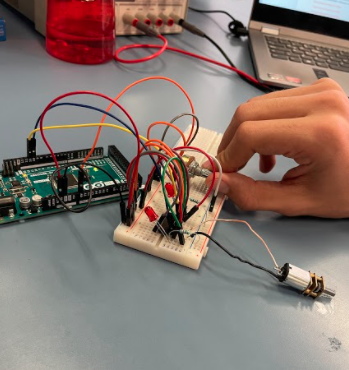
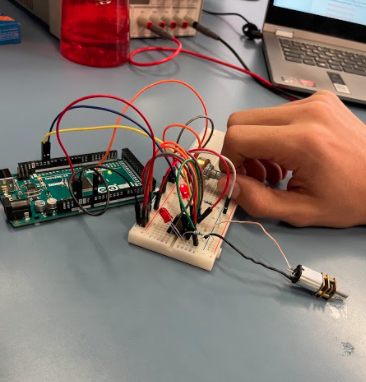
**Serial.print(MotA);**

**Serial.println(MotB);**

**delay(1000);**

**}**

****

****

* 1. **No se realizó**
  2. **Actividad 4 – Control proporcional de la velocidad**

*Tabla 3. Tabla de estados 2.*

| **Estado** | **Voltaje del potenciómetro** | **Acción del motor** |
| --- | --- | --- |
| 1 | 0 V |  |
| 2 | 0 V a 2 V | Valores intermedios de |
| 3 | 2 V |  |
| 4 | 2 V a 3 V |  |
| 5 | 3V |  |
| 6 | 3 V a 5 V | Valores intermedios de |
| 7 | 5V |  |

**//Act4**

**#define MotA 10 //se definen pines**

**#define MotB 11**

**#define A0 A0**

**void setup() {**

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

**pinMode(MotA, OUTPUT); // pin del led rojo1 como salida**

**pinMode(MotB, OUTPUT); // pin del led rojo2 como salida**

**Serial.begin(9600);**

**}**

**void loop() {**

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

**// se lee el voltaje del potenciómetro y se convierte a voltaje**

**float x = analogRead(A0)/204.6;**

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

**Serial.println(x);**

**// mapea el voltaje de [0,5]V a [0,255] para aumentar o disminuir la potencia del movimiento del motor.**

**int y = map(x\*1000, 0, 5000, 0, 255);**

**// Se hace el giro cómo se muestra en la tabla. Usamos freno pasivo. El parámetro ‘y’ es la intensidad de movimiento con la que se**

**// se mueve el motor.**

**if(x >= 0 && x < 2){**

**// Gira a la izquierda**

**levogiro(y);**

**} if (x >= 2 && x < 3){**

**// frena**

**frenoP(y);**

**} if (x >= 3 && x <= 5){**

**// Gira a la derecha**

**dextrogiro(y);**

**}**

**}**

**void levogiro(int y){**

**//Mandamos 1 y 0**

**analogWrite(MotA, 255 - y);**

**digitalWrite(MotB, LOW);**

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

**Serial.print(MotA);**

**Serial.println(MotB);**

**}**

**void frenoP(){**

**//Mandamos 0 y 0**

**digitalWrite(MotA, LOW);**

**digitalWrite(MotB, LOW);**

**Serial.print("Freno pasivo: ");**

**Serial.print(MotA);**

**Serial.println(MotB);**

**}**

**void dextrogiro(int y){**

**//Mandamos 0 y 1**

**digitalWrite(MotA, LOW);**

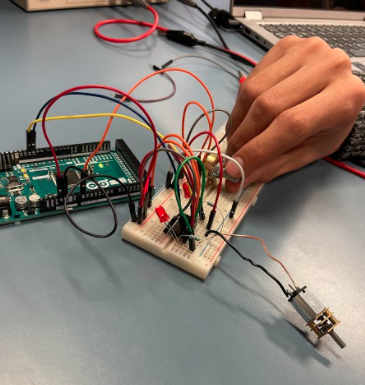
**analogWrite(MotB, y);**

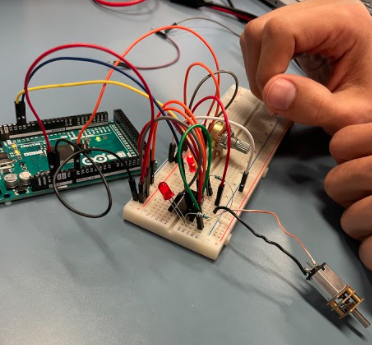
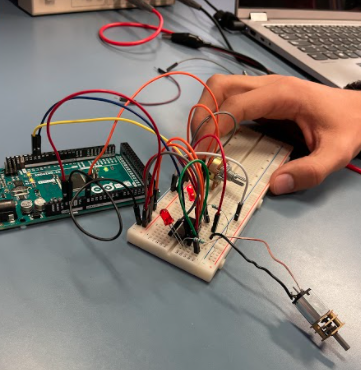
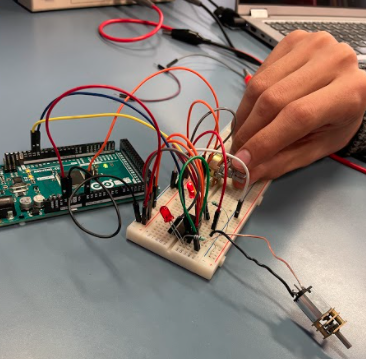
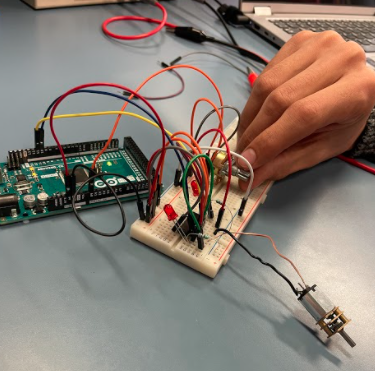
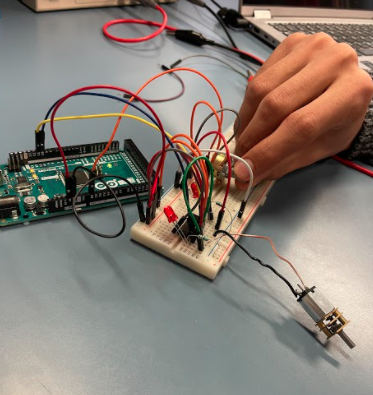
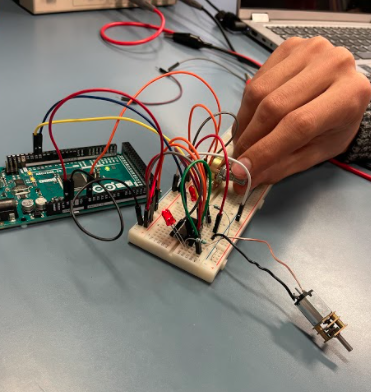
**Serial.print("dextrogiro: ");**

**Serial.print(MotA);**

**Serial.println(MotB);**

**}**

****

****

1. **Conclusiones**

Esta práctica nos fue muy útil porque aprendimos cómo funciona el motor DC con escobillas. Esto nos va a permitir utilizar diferentes tipos de motores para las próximas prácticas. Además, con los estados, podemos definir los movimientos que el motor debe hacer. Por último, gracias al conocimiento adquirido, estamos un paso más cerca de poder diseñar el robot que será capaz de moverse automáticamente hacia la luz.

1. **Referencias**

Colaboradores de los proyectos Wikimedia. (2008, March 10). *Puente H (electrónica)*. Wikipedia.org; Wikimedia Foundation, Inc. <https://es.wikipedia.org/wiki/Puente_H_(electr%C3%B3nica)>

Dice, S. (2017, November 7). *L293D - Controlador de motores*. HETPRO/TUTORIALES. <https://hetpro-store.com/TUTORIALES/l293d/>

*Diferencias entre motores con escobillas y brushless. (2016, November 29).* [*https://clr.es/blog/es/diferencias-motores-con-escobillas-brushless/#:~:text=Los%20motores%20con%20escobillas%2C%20como*](https://clr.es/blog/es/diferencias-motores-con-escobillas-brushless/#:~:text=Los%20motores%20con%20escobillas%2C%20como)

*Motor paso a paso*. (2021, January 23). Wikipedia. <https://es.wikipedia.org/wiki/Motor_paso_a_paso>

‌

‌