	INFORME DE LABORATORIO		octubre 18 de 2024
			Versión: 002
	<i>Electiva II y Electrónica del computador</i>		Página 1 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa

INTEGRANTES:

-Isaac Contreras Padilla

-Fabian Carcamo Jimenez

-Omar David Fuertes Garcia

-Yeremys Sierra Miranda

1. Datos del Autor

Datos del docente Orientador	<i>Miguel Alberto Caro Álvarez</i>
Programa	<i>TECNOLOGÍA DE DESARROLLO DE SISTEMAS DE INFORMACIÓN Y DE SOFTWARE</i>
Email	<i>Miguel.caro@curnvirtual.edu.co</i>

2. Generalidades

Lo más importante a destacar de este proyecto es su finalidad: garantizar que opere correctamente aplicando las reglas principales. Además, el uso de Arduino como herramienta para su desarrollo nos impulsa a practicar continuamente, fomentando la creación de diversos proyectos.


3. Objetivos de la práctica

El objetivo de este proyecto es crear y ejecutar varios mecanismos para circuitos simples. Usando un Arduino y C++ para ejecutar este proyecto, una de las principales características comunes de este proyecto es el uso de varias herramientas de ejecución y sangría.

4. Marco teórico

¿Qué son los sensores?

Según la Real Academia Española, un sensor es un dispositivo que detecta una determinada acción externa, temperatura, presión, etcétera, y la transmite adecuadamente. Este artefacto toma las señales analógicas externas y las convierte en impulsos eléctricos que son capturados por una serie de circuitos, los cuales modifican para realizar cierto tipo de comportamientos. Estos dispositivos reaccionan a los cambios de las condiciones físicas alterando sus propiedades eléctricas. Por lo tanto, la mayoría de estos dispositivos industriales dependen de sistemas electrónicos para capturar, analizar y transmitir información sobre el medio ambiente.

	INFORME DE LABORATORIO		Octubre 18 de 2024
			Versión: 002
	<i>Electiva II y Electrónica del computador</i>		Página 2 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa

Las características principales de los sensores son:


- **Rango.** Es el valor mínimo y máximo de la variable física que el sensor puede percibir o medir.
- **Amplitud.** Es la diferencia entre los valores máximos y mínimos de entrada.
- **Exactitud.** El error en la medición se especifica en términos de precisión. Se define como la diferencia entre el valor medido y el valor real. Se define en términos de % de la escala completa o % de la lectura.
- **Precisión.** Se define como la cercanía entre un conjunto de valores y es diferente de la exactitud.
- **Sensibilidad.** Es la relación entre el valor de la salida y el valor de la entrada.
- **La alineación.** Es la máxima desviación entre los valores medidos de un sensor de la curva ideal.
- **Histéresis.** Es la diferencia en la salida cuando la entrada varía de dos maneras, aumentando y disminuyendo.
- **Resolución.** Es el cambio mínimo en la entrada que puede ser detectado por el sensor.
- **Reproducibilidad.** Se define como la capacidad del sensor de producir la misma salida cuando se aplica la misma entrada.
- **Repetibilidad.** Capacidad del sensor de producir la misma salida cada vez que se aplica la misma entrada y todas las condiciones físicas y de medición se mantienen iguales, incluyendo el operador, el instrumento, las condiciones ambientales, etc.
- **Tiempo de respuesta.** Se expresa generalmente como el tiempo en que la salida alcanza un cierto porcentaje de su valor final, en respuesta a un cambio de paso de la entrada.

Sensores de proximidad

Son transductores que detectan objetos o señales que se encuentran cerca del elemento sensor. Existen varios tipos de sensores de proximidad según el principio físico que utilizan, los más comunes son los inductivos, magnéticos, capacitivos, ultrasónicos.

¿Qué es arduino?

Es una herramienta utilizada en la programación, una placa electrónica de código abierto o hardware libre que se utiliza para las conexiones de microcontroladores y los diferentes sensores que hacen parte de un circuito donde permite escribir, cargar y ejecutar código en la placa, las placas Arduino incluyen entrada y salida digitales y analógicos que permiten la conexión y control de otros dispositivos

	INFORME DE LABORATORIO		octubre 18 de 2024
			Versión: 002
	<i>Electiva II y Electrónica del computador</i>		Página 1 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa

electrónicos. Para comprender su funcionamiento, es importante conocer sus componentes principales:


- A. Microcontrolador: Es el cerebro del Arduino. Está formado por un circuito integrado capaz de almacenar y ejecutar el programa que le cargamos. El más común es el ATmega328.
- B. Interruptor de Encendido (Botón de Reset): Permite activar o desactivar la placa. Al presionarlo, reiniciamos el programa.
- C. Entradas Analógicas y Digitales
- D. Entradas Analógicas: Toman indicaciones directamente desde sensores y miden valores continuos (como temperatura o luz).
- E. Entradas digitales: Reciben datos binarios (encendido/apagado) y se utilizan para leer señales digitales (como las de los LEDs o los botones).
- F. Conector USB: Nos permite conectar la placa Arduino al ordenador para cargar el código.
- G. LED (Diodos emisores de luz): Están ubicados en la placa y se iluminan cuando recibe electricidad. Sirven para verificar si la placa está funcionando correctamente.
- H. Terminales de salida: Reciben la información procesada por el Arduino y la transmiten a otros dispositivos externos, como motores, relés o pantallas.

El proceso de funcionamiento del Arduino es el siguiente:

1. El usuario escribe un programa en el lenguaje de programación "Arduino" y lo guarda en el computador.
2. Luego, carga el programa en el Arduino mediante el puerto USB.
3. El microcontrolador "lee" las señales de entrada y salida a través de sus interruptores de encendido/apagado.
4. Procesa los datos según el código del programa.
5. Los resultados se envían a través de los pines de salida, conectando el Arduino con otros dispositivos externos.

Diseño de circuito

El diseño de circuitos electrónicos es una fascinante área de la electrónica que abarca la creación de circuitos tanto analógicos como digitales. Este proceso involucra diversas etapas, desde la especificación inicial hasta la verificación y prueba del diseño final. Aunque puede parecer complejo, con la orientación adecuada es posible lograrlo con éxito. Cada fase, desde la concepción hasta la implementación, es crucial para obtener un resultado funcional y eficiente en la manipulación de señales eléctricas hacia un objetivo específico.

	INFORME DE LABORATORIO		Octubre 18 de 2024
			Versión: 002
	<i>Electiva II y Electrónica del computador</i>		Página 4 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa

Un circuito electrónico incluye componentes como resistencias, condensadores, diodos y transistores conectados por cables, por donde fluye la corriente. Se diseña inicialmente en una placa de pruebas (prototipos) para permitir modificaciones y mejoras por parte del diseñador.


Los componentes para diseñar un circuito electrónico son:

- Transistores:** Son dispositivos semiconductores que tienen la capacidad de amplificar o conmutar señales electrónicas. Representado por tres líneas que forman un triángulo.
- Resistencias:** Son elementos pasivos que se utilizan para limitar el flujo de corriente en un circuito. Representada por un rectángulo con una línea ondulada en su interior.
- Capacitores:** Son componentes que almacenan energía en forma de carga eléctrica. Estos elementos se utilizan para filtrar señales, generar temporizadores y almacenar energía en circuitos de alimentación. Está representado por dos líneas paralelas y una línea curva entre ellas.
- Inductores:** Son elementos que almacenan energía en forma de campo magnético. Estos componentes se utilizan principalmente en circuitos de corriente alterna y son fundamentales en aplicaciones de potencia y filtrado. Los inductores están compuestos por un núcleo magnético y una bobina de alambre. Representado por una especie de bobina.
- Diodos:** Los diodos son componentes semiconductores que permiten el flujo de corriente en una dirección y bloquean el flujo en la opuesta. Representado por una línea con una flecha en uno de sus extremos.
- Herramientas de medición:** Es importante contar con herramientas de medición precisas para diseñar y evaluar circuitos electrónicos. Algunas de las herramientas más comunes son el multímetro, el osciloscopio y el generador de señales.

Además de los símbolos, es fundamental comprender también la terminología utilizada en los diagramas de circuitos.

Es fundamental entender los conceptos básicos de voltaje, corriente y resistencia.

- El voltaje es una magnitud física que da cuenta de la diferencia en el potencial eléctrico entre dos puntos determinados.
- La corriente es el flujo de carga eléctrica a través de un material conductor. Esta corriente es esencial en nuestra vida diaria y se encuentra en prácticamente todos los dispositivos electrónicos que utilizamos.
- La resistencia es una medida de la oposición al flujo de corriente en un circuito eléctrico.

	INFORME DE LABORATORIO		octubre 18 de 2024
			Versión: 002
	<i>Electiva II y Electrónica del computador</i>		Página 1 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa

¿Qué es la Librería PID_v1?

Esta librería simplifica la implementación del algoritmo PID y es una herramienta de software utilizada para la implementación de controladores PID (Proporcional-Integral-Derivativo) en sistemas de control. Estos controladores son ampliamente utilizados en la ingeniería para regular variables como temperatura, velocidad, posición, etc., en una amplia gama de aplicaciones industriales y de automatización.

¿Qué es la librería Wire.h?

permite la comunicación I2C (Inter-Integrated Circuit) entre el microcontrolador y otros dispositivos. I2C es un protocolo de comunicación serial que utiliza dos líneas (SDA para datos y SCL para el reloj) para la transmisión de datos entre múltiples dispositivos.

¿Qué es la librería Servo.h?

permite el control de servomotores. Los servos son motores que permiten el control preciso de la posición angular, muy usados en robótica y automatización.

¿Qué es la librería MPU6050.h?

permite la comunicación con el sensor de movimiento MPU6050, que incluye un acelerómetro y un giroscopio. Este sensor se utiliza para medir la aceleración y la velocidad angular en tres ejes.

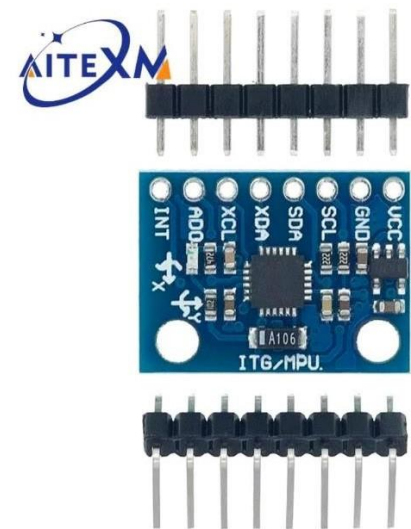
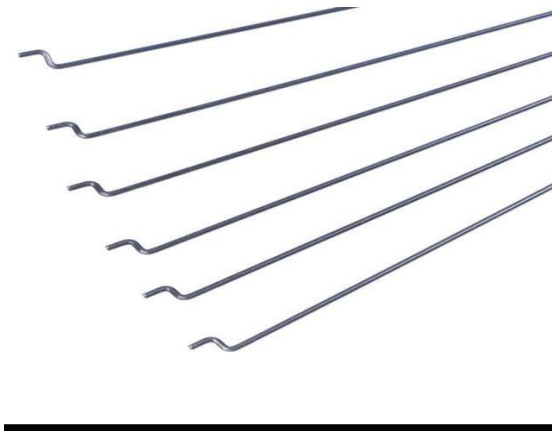
5. Metodología

5.1 Materiales y equipos

- * Placa Protoboard
- * Arduino uno
- * Sensor mpu 6050
- * Servo 9g
- * Clavijas de avión de RC
- * Varilla de empuje
- * Bloque de poliestireno expandido



	Informe de Laboratorio		Octubre 18 de 2024
			Versión: 002
	Electiva II y Electrónica del computador		Página 6 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa



	INFORME DE LABORATORIO	octubre 18 de 2024
		Versión: 002
	<i>Electiva II y Electrónica del computador</i>	Página 1 de 7
	Ajuste automático de los alerones de un avión	Aprobado por: Ing. Yenny Julio Director de Programa



5.2 Procedimiento

Lenguaje en Arduino

```
aleron.ino
1 #include <Wire.h>
2 #include <PID_v1.h>
3 #include <Servo.h>
4 #include <MPU6050.h>
5
6 // Inicializar el MPU6050 y el servo
7 MPU6050 mpu;
8 Servo aleronServo;
9
10 // Definir variables para el PID
11 double setpoint, input, output;
12 double Kp = 2, Ki = 5, Kd = 1; // Coeficientes PID ajustables
13
14 // Crear objeto PID
15 PID myPID(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);
16
17 void setup() {
18   // Iniciar la comunicación I2C
19   Wire.begin();
20
21   // Iniciar el puerto serie para depuración
22   Serial.begin(9600);
23
24   // Inicializar el sensor MPU6050
25   mpu.initialize();
26
27   // Verificar si el MPU6050 está conectado correctamente
28   if (!mpu.testConnection()) {
29     Serial.println("Error al conectar el MPU6050");
30     while (1);
31   }
32
33   // Inicializar el servomotor (controlará ambos alerones)
34   aleronServo.attach(9); // Pin digital 9 para el servo
35
36   // Definir el setpoint como 0 grados de inclinación (estabilidad)
37 }
```

Output
El sketch usa 9930 bytes (30%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.

	INFORME DE LABORATORIO		Octubre 18 de 2024
			Versión: 002
	<i>Electiva II y Electrónica del computador</i>		Página 8 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa

```
aleron.ino
34 // Definir el pin al que se conecta el servo
35
36 // Definir el setpoint como 0 grados de inclinación (estabilidad)
37 Setpoint = 0;
38
39 // Configurar el PID
40 myPID.SetMode(AUTOMATIC);
41 myPID.SetOutputLimits(0, 180); // limitar la salida para el servo
42
43 Serial.println("Sistema listo");
44 }
45
46 void loop() {
47 // Leer datos de inclinación (giroscopio o acelerómetro)
48 int16_t ax, ay, az;
49 mpu.getAcceleration(&ax, &ay, &az);
50
51 // Calcular la inclinación en grados a partir de los datos del acelerómetro
52 Input = atan2(ay, az) * 180 / PI;
53
54 // Ejecutar el cálculo del PID
55 myPID.Compute();
56
57 // Mover el servomotor según la salida del PID (controla ambos alerones)
58 aleronServo.write(Output);
59
60 // Mostrar datos para depuración
61 Serial.print("Inclinación: ");
62 Serial.print(Input);
63 Serial.print(" | Salida Servo (Alerones): ");
64 Serial.println(Output);
65
66 // Pequeño retraso
67 delay(50);
68 }
69
```


Output

El Sketch usa 9930 bytes (30%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.

Ln 4, Col 21 Arduino Uno on COM6

Explicación:

- 1. **Inclusión de Librerías:**
 - Wire.h: Para la comunicación I2C.
 - PID_v1.h: Para el controlador PID.
 - Servo.h: Para controlar el servomotor.
 - MPU6050.h: Para interactuar con el sensor de movimiento MPU6050.
- 2. **Inicialización de Variables y Objetos:**
 - Se crean objetos mpu para el sensor MPU6050 y aleronServo para el servomotor.
 - Se declaran las variables Setpoint, Input y Output para el controlador PID.
 - Se definen las constantes PID Kp, Ki, Kd.
 - Se inicializa el objeto PID con las variables y constantes definidas.
- 3. **Configuración Inicial (setup):**
 - Se inicia la comunicación I2C con Wire.begin().
 - Se inicia la comunicación serie para la depuración con Serial.begin(9600).
 - Se inicializa el sensor MPU6050 con mpu.initialize().
 - Se verifica la conexión del MPU6050 con mpu.testConnection(). Si falla, se imprime un mensaje de error y el programa se detiene.

	INFORME DE LABORATORIO		octubre 18 de 2024
			Versión: 002
	<i>Electiva II y Electrónica del computador</i>		Página 1 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa

- Se configura el servomotor en el pin 9 con `aleronServo.attach(9)`.
- Se establece el punto de ajuste (Setpoint) en 0 grados.
- Se configura el PID en modo automático y se limitan las salidas del PID al rango del servomotor (0 a 180 grados).

4. Bucle Principal (loop):

- Se leen los datos de aceleración del MPU6050 con `mpu.getAcceleration(&ax, &ay, &az)`.
- Se calcula la inclinación en grados utilizando la función `atan2` y se convierte de radianes a grados.
- Se ejecuta el cálculo del PID con `myPID.Compute()`.
- Se ajusta el servomotor según la salida del PID con `aleronServo.write(Output)`.
- Se imprimen los valores de inclinación y la salida del servo para depuración.
- Se introduce un pequeño retraso de 50 ms con `delay(50)`.

Lenguaje de programación C++

```
#include <PID_v1.h>

// Define los pines de entrada y salida
const int pinSensor = A0; // Pin analógico para leer la posición actual
const int pinServo = 9;   // Pin digital para controlar el servomotor

// Define los parámetros del PID
double Setpoint, Input, Output;
PID myPID(&Input, &Output, &Setpoint, 2, 5, 1, DIRECT);

void setup() {
    // Inicializa el sensor y el servo
    pinMode(pinServo, OUTPUT);
    myPID.SetMode(AUTOMATIC);
    Setpoint = 90; // Ángulo de referencia del alerón (ajusta según tus ne
}

void loop() {
    // Lee la posición actual del alerón desde el sensor
    Input = analogRead(pinSensor);

    // Calcula la salida del PID
    myPID.Compute();

    // Limita la salida del PID para evitar valores fuera de rango
    Output = constrain(Output, 0, 180);

    // Escribe la señal de control al servomotor
    analogWrite(pinServo, Output);

    // Espera un tiempo antes de la siguiente iteración
    delay(10);
}
```

	INFORME DE LABORATORIO	Octubre 18 de 2024
		Versión: 002
	<i>Electiva II y Electrónica del computador</i>	Página 10 de 7
	Ajuste automático de los alerones de un avión	Aprobado por: Ing. Yenny Julio Director de Programa

Explicación:

Librería PID_v1: Esta librería simplifica la implementación del algoritmo PID.

Pines: Se definen los pines del Arduino que se utilizarán para leer la posición del alerón y controlar el servomotor.

Parámetros PID:


- **Setpoint:** Ángulo de referencia deseado para el alerón.
- **Input:** Valor leído del sensor (posición actual del alerón).
- **Output:** Señal de control enviada al servomotor.
- **2, 5, 1:** Valores iniciales para las constantes Kp, Ki y Kd. Estos valores deben ajustarse experimentalmente para obtener un buen desempeño del control.

Setup:

- Se inicializa el sensor y el servomotor.
- Se establece el modo automático del PID y el valor del Setpoint.

Loop:

- Lee la posición actual del alerón.
- Calcula la salida del PID utilizando la función **myPID.Compute()**.
- Limita la salida del PID para que esté dentro del rango válido del servomotor (0-180 grados).
- Escribe la señal de control al servomotor utilizando **analogWrite()**.

	INFORME DE LABORATORIO		octubre 18 de 2024
			Versión: 002
	<i>Electiva II y Electrónica del computador</i>		Página 1 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa

5.3 Precauciones o riesgos de la práctica

1. Conexiones Eléctricas

- Revisar las Conexiones: Asegúrate de que todas las conexiones eléctricas sean firmes y correctas. Las conexiones sueltas pueden causar un funcionamiento inestable o incluso dañar los componentes.
- Aislamiento: Usa aislamiento adecuado en las conexiones para evitar cortocircuitos.
- Voltajes Correctos: Verifica que los voltajes de operación sean adecuados para todos los componentes. Por ejemplo, el MPU6050 generalmente funciona a 3.3V, mientras que los servomotores pueden requerir 5V.

2. Alimentación

- Fuente de Alimentación: Usa una fuente de alimentación estable y adecuada para el Arduino y los servomotores. Los servomotores pueden requerir más corriente de la que puede proporcionar el Arduino, por lo que puede ser necesario un suministro de energía externo.
- Condensadores de Desacoplo: Considera agregar condensadores de desacoplo cerca de los pines de alimentación del MPU6050 y del servomotor para reducir el ruido eléctrico.

3. Configuración del PID

- Tuning del PID: Los valores de Kp, Ki y Kd pueden necesitar ajustes para obtener una respuesta adecuada del sistema. Un mal ajuste puede provocar oscilaciones, respuesta lenta o inestabilidad.
- Pruebas Iniciales: Realiza pruebas iniciales con los servomotores sin cargas pesadas para ajustar los parámetros PID sin riesgo de daño.

4. Mecánica del Sistema


- Montaje Seguro: Asegúrate de que el MPU6050 esté firmemente montado en la estructura para evitar lecturas inestables debido a vibraciones.
- Servomotor: Verifica que el servomotor pueda moverse libremente sin obstrucciones y que esté adecuadamente sujeto.

5. Código y Lógica

- Verificación de Errores: Añade más verificaciones de errores y mensajes de depuración para identificar problemas fácilmente.
- Tiempo de Ejecución: Asegúrate de que el delay(50) en el bucle principal no afecte negativamente la respuesta del sistema. En aplicaciones críticas, podrías usar temporizadores en lugar de delay.

6. Seguridad Personal

- Evitar Contacto Directo: No toques el servomotor ni los componentes electrónicos mientras el sistema esté en funcionamiento.
- Uso de Protección: Considera usar gafas de seguridad y otras protecciones si estás realizando pruebas mecánicas con partes móviles.

	INFORME DE LABORATORIO		Noviembre 1 de 2024
	<i>Electiva II y Electrónica del computador</i>		Versión: 003
	Ajuste automático de los alerones de un avión		Página 12 de 7
			Aprobado por: Ing. Yenny Julio Director de Programa

7. Pruebas

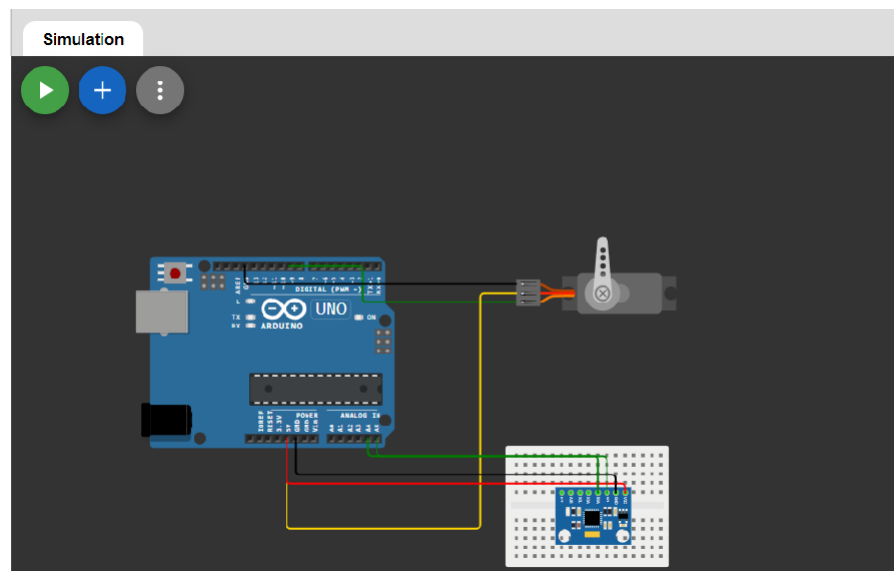
- Pruebas en Entorno Controlado: Realiza pruebas iniciales en un entorno controlado donde cualquier fallo no cause daños a personas o equipos.
- Monitorización Constante: Durante las pruebas, monitorea constantemente los valores de inclinación y las respuestas del servomotor.


8. Documentación y Planificación

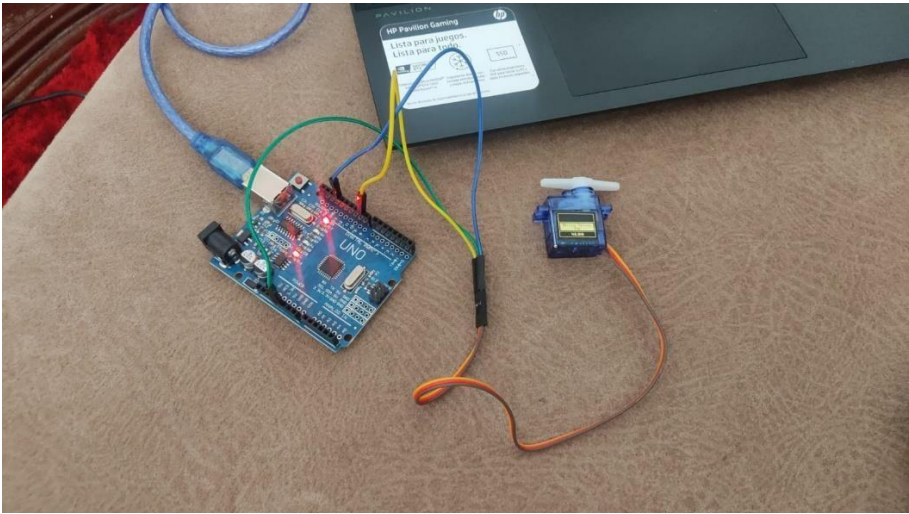
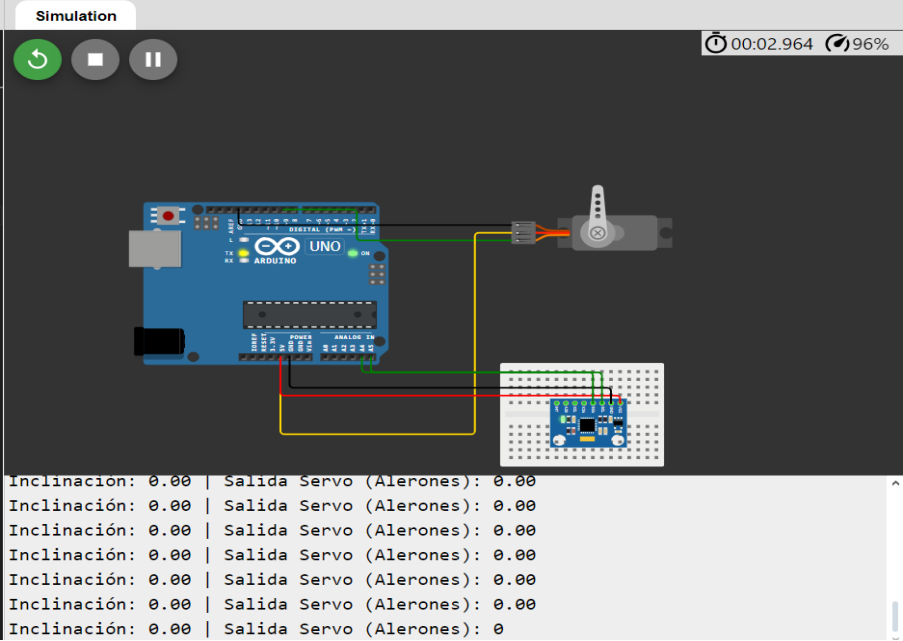
- Documentar Cambios: Documenta cualquier cambio en el código y las configuraciones de hardware.
- Planificación de Fallos: Planifica cómo manejar fallos del sistema, como pérdida de señal del sensor o fallos en el servomotor.

6. Resultados y conclusiones de la práctica

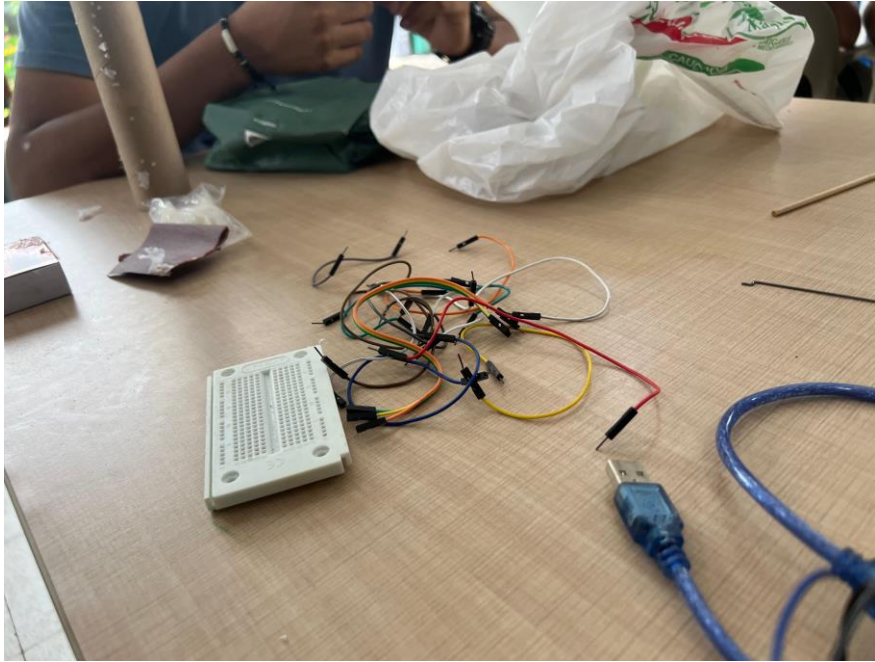
Link de prueba: <https://wokwi.com/projects/412088088398023681>



	INFORME DE LABORATORIO	Noviembre1 de 2024
		Versión: 002
	<i>Electiva II y Electrónica del computador</i>	Página 1 de 7
	Ajuste automático de los alerones de un avión	Aprobado por: Ing. Yenny Julio Director de Programa




	INFORME DE LABORATORIO		Noviembre 1 de 2024
	<i>Electiva II y Electrónica del computador</i>		Versión: 003
	Ajuste automático de los alerones de un avión		Página 14 de 7
			Aprobado por: Ing. Yenny Julio Director de Programa



	INFORME DE LABORATORIO		Noviembre1 de 2024
			Versión: 002
	<i>Electiva II y Electrónica del computador</i>		Página 1 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa



	INFORME DE LABORATORIO		Noviembre 1 de 2024
			Versión: 003
	<i>Electiva II y Electrónica del computador</i>		Página 16 de 7
	Ajuste automático de los alerones de un avión		Aprobado por: Ing. Yenny Julio Director de Programa

7. Referencias bibliográficas

<https://sdindustrial.com.mx/blog/sensores/>

https://www.elespanol.com/invertia/disruptores-innovadores/innovadores/centros-tecnologicos/20210127/trenes-aterrizaje-inteligentes-gracias-sensores-opticos-invisibles/552695720_0.html

<https://descubrearduino.com/puertas-logicas/>

<https://electrositio.com/diferentes-procesos-de-diseno-de-circuitos-electronicos/>

<https://es.slideshare.net/mar97/tipos-de-circuitos-electrnicos-13050622>

<https://gerocastillo.com/electronica/disenar-circuito-electronico-concepto-implementacion/>

8. Control de Actualizaciones

Versión anterior	Fecha del Documento Anterior	Descripción del Cambio
Versión 001	28 de agosto de 2024	Se actualiza, primera entrega
Versión 002	18 de octubre de 2024	Se actualiza, segunda entrega
Versión 003	1 de noviembre de 2024	Se actualiza, última entrega