Laboratorio 2

1st Belen Boado
Ingenieria Mecatronica
Universidad Tecnologica (UTEC)
Fray Bentos, Rio Negro
belen.boado@estudiantes.utec.edu.uy

2nd Luis Nuñez
Ingenieria Mecatronica
Universidad Tecnologica (UTEC)
Fray Bentos, Rio Negro
luis.nunez@estudiantes.utec.edu.uy

3rd Adrian Terzaghi
Ingenieria Mecatronica
Universidad Tecnologica (UTEC)
Fray Bentos, Rio Negro
adrian.terzaghi@estudiantes.utec.edu.uy

***Index Terms*—ATmega328P, LDR, Potenciometro, Plotter, Planta**

I. Resumen

En esta práctica de laboratorio , se desarrollaron 4 problemas. Los cuales se detallaran a continuación. Como primer problema tenemos el desarrollo de un código en c para el control de un plotter, teniendo en cuenta que tiene que ser mediante comunicación serial , permitiendo al usuario seleccionar y poder dibujar diversas figuras.Cumpliendo con ciertas características como lo son ,el mostrar un menú interactivo a través del puerto serial ,dibujar las figuras predefinidas: triángulo, círculo , cruz, perro y conejo..El ejecutar las secuencias de dibujo considerando los tiempos de conmutación de los relés y garantizando precisión en el movimiento del plotter y la edición personalizada de los dibujos por parte del usuario. Para el segundo problema el microcontrolador debe ser capaz de medir la temperatura a través de una PT100 cada 5 segundos. Y dependiendo de la temperatura tomar acciones como prender o apagar el ventilador y lámparas. Debe de comunicarse por puerto serie e informar en el estado que se encuentra e indicar la temperatura. Problema 3 el microcontrolador debe ser capaz de leer el valor de un potenciómetro y regular otro potenciómetro que quede de igual valor a través del control de un motor que está conectado al potenciómetro. Informando el sentido de giro y mostrando una grafica. Como último problema , nos encontramos con hacer un sistema que detecte el color al que se apunta a través de una fotocelda y ser capaz de indicar que color es atravez del control de un servo motor.

II. Introducción

Este informe presenta una serie de experimentos realizados con el microcontrolador ATmega328P, enfocados en el diseño e implementación de sistemas que combinan sensores, actuadores y comunicación serial. Los ejercicios abarcan desde el control de motores y servomotores hasta la obtencion de datos de sensores analógicos.

Entre las actividades realizadas, se incluyeron sistemas de control basados en la modulación por ancho de pulso (PWM) para ajustar la velocidad y dirección de un motor, y el uso de potenciómetros para establecer referencias de control. Además, se trabajó con sensores como la fotoresistencia (LDR) y el sensor de temperatura PT100, permitiendo no solo la medición de variables físicas, sino también su procesamiento y visualización en tiempo real a través de la comunicación UART.

Asimismo, se implementaron sistemas de control de precisión, como en el caso de un plotter para realizar dibujos, donde se coordinó el movimiento de motores paso a paso, y en otro caso, la lectura de sensores para ajustar automáticamente la respuesta de actuadores, como servomotores y motores DC.

A continuacion se detallan los objetivos, metodologías y resultados obtenidos en cada experimento, proporcionando conocimientos acerca del funcionamiento de algunos procesos fisicos y como realizar monitoreos en tiempo real de variables.

III. Objetivos

A. Objetivo General

Lograr los correctos funcionamientos de los codigos, simulaciones y montajes fisico de los problemas planteados en la guia del informe.

B. Objetivos especificos

- Implementacion de codigo y dibujo en un plotter.
- Implementacion del codigo, simulacion y montaje fisico de la planta simuladora de cambio de temperatura a base del sensor PT100.
- Implementacion de codigo, simulacion y montaje fisico de dos potenciometros y un motor DC.
- Implementacion de codigo, simulacion y montaje fisico de un circuito para cambio de color con una fotoresistencia y un servomotor manejado por PWM.
- Visualizacion de datos a traves de la libreria pyserial.

IV. Marco teórico

A continuacion, se plantearan algunos conceptos fundamentales para la realizacion de esta practica, como el concepto del microcontrolador ATmega328P, una breve definicion del lenguaje C, y algunas definiciones de componentes importantes de los problemas a realizar.

A. Microcontrolador ATmega328P

El microcontrolador ATmega328 es un microcontrolador de 8 bits basado en una arquitectura de RISC. Posee muchas características como una memoria flash de 32KB con una capacidad de lectura en simultáneo a escritura, una memoria EEPROM de 1KB, un SRAM de 2KB y 23 pines de uso.

Además, posee 32 registros, 3 temporizadores flexibles, soporta interrupciones externas y posee internas, una interfaz para la comunicación serial en los pines PD0 y PD1, y Conversor Análogo Digital. Por último, posee temporizador de modo de ahorro de energía y un temporizador "Watchdog". Este microcontrolador opera entre 1.8 y 5.5 volts.

Un reemplazo frecuente para este microcontrolador es el ATmega328P, ya que comparten en su mayoría las mismas características. Este se encuentra comúnmente en la plataforma de Arduino UNO y Arduino Nano.

1) Caracteristicas:

- 131 instrucciones.
- Hasta 20 MIPS a 20 MHz.
- Memoria FLASH programable en el sistema de 32 KB.
- 2 KB de SRAM interna.
- 1 KB de EEPROM.
- 6 canales PWM.
- 6 canales analógicos para el ADC.
- Puerto USART, SPI e interfaz serial de 2 cables.
- Temporizador watchdog y comparador analógico.
- Soporte para múltiples modos de bajo consumo.

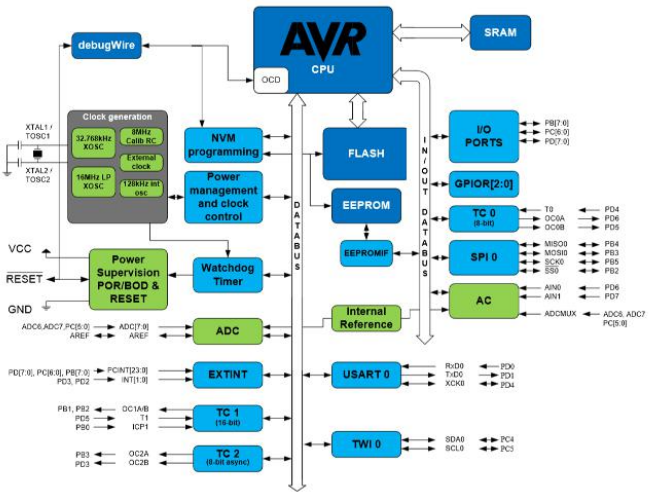

Fig. 1. Circuito interno del ATmega328P.

B. Lenguaje C

Este lenguaje es uno de los más importantes y utilizados a nivel informático. Fue desarrollado en la década de 1970 por los Laboratorios Bell por Dennis Ritchie, evolucionando del lenguaje B y fue creado con la idea de implementar el sistema operativo UNIX, por lo cual fue muy popular en la industria.

C es un lenguaje destacado por su capacidad de manipulación de hardware y memoria, volviendolo un lenguaje eficiente y portatil.Su impacto ha sido muy grande en la industria, influyendo en la creación de lenguajes como C++, Java y Javascript.

1) Características:

- Manipulación directa de hardware y memoria.
- Eficiencia y portabilidad.
- Amplia influencia en otros lenguajes de programación.

C. Servomotores

Un servomotor es un motor diseñado para el control preciso de movimientos rotativos o lineales, utilizando un sistema de retroalimentación que asegura un posicionamiento exacto. Este tipo de motor se desarrolló a lo largo del siglo XX, inicialmente para aplicaciones en el ámbito militar y aeronáutico, y ha evolucionado para convertirse en un componente clave en la ingeniería moderna.

Su funcionamiento implica recibir una señal de control que indica la posición deseada y ajustar el movimiento del motor mediante un mecanismo de engranaje y un sistema de retroalimentación, que monitorea continuamente la posición actual. Esta capacidad de corrección de errores lo hace esencial en aplicaciones que requieren alta precisión, como la robótica y la maquinaria CNC.

Fig. 2. Servomotor.

D. Plotter

Los plotters pueden dividirse en 3 grandes grupos, los plotters de impresión, los de corte y los mixtos. Los de

impresión son considerados impresoras de gran formato, son ideales para imprimir desde planos hasta fotografías o carteles de gran tamaño. Los de corte se utilizan para cortar vinilos de colores o formas previamente impresas, y los mixtos combinan ambas funciones.

El plotter de impresión es una de las formas rápidas y eficientes de producir impresiones de gran tamaño de alta calidad. Además, con esta tecnología es posible realizar impresiones sobre tela, crear carteles publicitarios, planos de arquitectura e ingeniería.

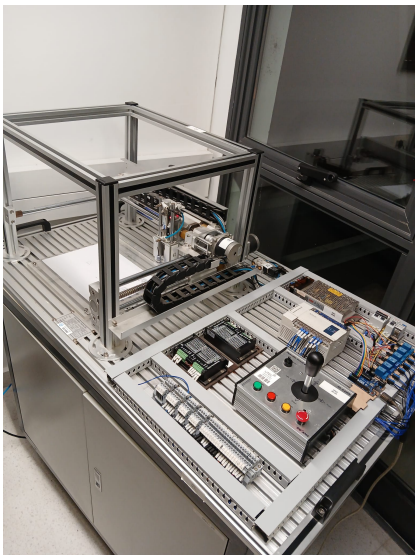

Fig. 3. Plotter del Laboratorio de Mecatronica.

E. Sensor PT100

El sensor de temperatura PT100 utiliza un RTD, o Detector de Temperatura por Resistencia, con el fin de medir distintos valores de temperatura. Está fabricado en platino, de allí su nombre, y tiene una resistencia de 100 Ω a cero grados centígrados. Es altamente utilizado por su precisión y amplios rangos de temperatura.

Este elemento es esencial para realizar ediciones de forma precisa en una gran variedad de campos, ya que sus rangos varían entre los -200 grados a los 850 grados.

Fig. 4. Sensor PT100.

F. Puente H

El integrado Puente H es un controlador de motores de corriente directa, el cual permite enviar señales a al menos 2 motores al mismo tiempo, proporcionando el control de giro independiente para cada uno.

Este integrado permite utilizar motores de un voltaje inferior a 36 voltios (según el modelo de integrado, en nuestro caso el máximo permitido es 12V) y una corriente proporcional a este valor.

El Puente H posee una configuración que permite que el voltaje fluya en cualquier dirección, se utiliza, generalmente, para hacer funcionar motores en sentido horario o antihorario.

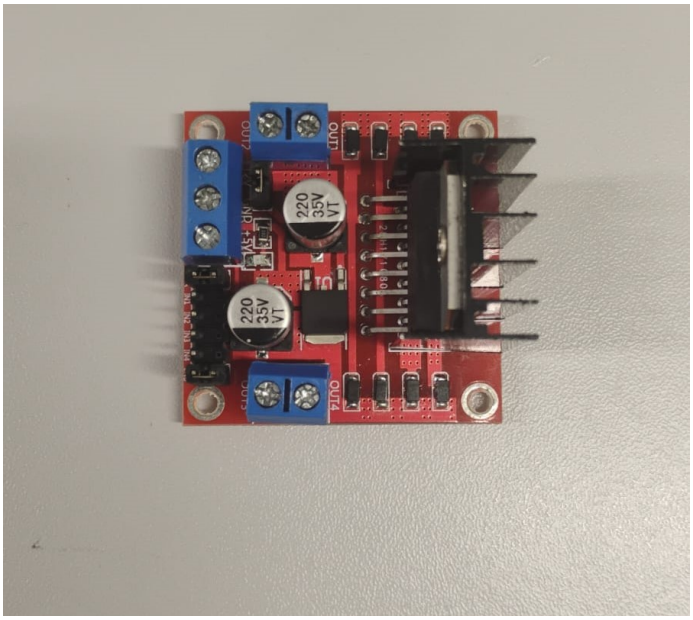

Fig. 5. Circuito integrado Puente H.

V. Metodología

A. Problema A

Para la resolución de este problema se utilizó el plotter de dibujo encontrado en el laboratorio de mecatrónica de la ITRSO de UTEC, el cual cuenta con un Arduino UNO conectado, que fue utilizado para controlarlo.

El objetivo de esta problemática fue realizar distintos dibujos de manera automática haciendo uso del microcontrolador integrado al arduino, así mismo se implementó el puerto serial con el objetivo de desplegar un menú de selección en el cual el usuario tiene la posibilidad de elegir el dibujo a realizar.

Con el fin de llevar a cabo los objetivos fue necesaria la utilización de bucles for para el dibujo de diagonales, teniendo en cuenta que era imposible accionar dos motores al mismo tiempo. Así mismo, se utilizó una técnica similar para la realización de curvas y del aproximado poligonal del círculo, donde se dibujaron distintas diagonales con diferentes pendientes para dar la ilusión de curvatura.

Para el dibujo de un perro y un conejo se optaron por diseños los cuales no presentan curvas, y que las figuras se vean representadas plenamente por líneas rectas.

B. Problema B

En está problema fue necesario el uso del microcontrolador ATmega328P, resistencias , amplificadores operacionales LM741,luces de 12v,ventilador de 12V ,relé y una PT100, todo esto con el fin de poder medir y controlar la temperatura dentro de un rango deseado en un ambiente aislado como lo es una caja. Para poder resolver este problema tenemos que medir la temperatura del ambiente , lo cual lo logramos a través de la PT100, esta nos brinda una resistencia variable dependiendo de la temperatura, para poder leer correctamente con el arduino necesitamos pasar esa resistencia variable a voltaje, para lo cual utilizamos un puente de Wheatstone, armado a través de dos resistencias de 1k ohm y una de 100 ohms ,logramos el cometido de pasar a voltaje . Al ser un voltaje muy pequeño , conectaremos a los amplificadores que estarán en una configuración de instrumentación, esto nos permitirá amplificar la señal y leer mejor el valor del voltaje. Tomando varios valores de voltaje y resistencia para diferentes temperaturas, realizamos una gráfica y determinamos la ecuación que nos dará la temperatura más adelante. Para el control se encargará el ATmega328P deberá leer el valor desde amplificador que viene desde la PT100 y procesar esa información , dependiendo del caso , prenderá el ventilador a una velocidad lenta o rápida , esto lo realizará con pwm(modulación por ancho de pulso) y también prender los focos en caso de temperaturas bajas, lo cual lo hará dando un uno o cero a través de su puerto que está comunicado al relé , que hará de interruptor con la lámparas de 12V . A todo esto ,se estará mostrando a través de puerto serie la temperatura medida y la acción que está tomando sea encender ventilador o no , igual para las lámparas encender o no. También el usuario tendrá la opción de ajustar los parámetros a través del puerto serie y poder ver en tiempo real la temperatura medida , gracias a usar Python para graficar los datos.

C. Problema C

Para este problema se utilizó el microcontrolador ATmega328P, dos potenciómetros de igual valor y un motor, todos estos elementos logran que a partir de un potenciómetro de referencia, se activa el motor y moverá el segundo potenciómetro a la posición más cercana al valor del de referencia. Para poder realizar estos movimientos, se emplea el uso de un PWM (modulación por ancho de pulso), que tomará los valores del potenciómetro de referencia como parámetro para variar la velocidad de giro del motor, ajustando el segundo potenciómetro.

Además, el sistema incorpora la comunicación serial para devolver el valor del potenciómetro de referencia, el valor del segundo potenciómetro, el del PWM y el sentido de giro del motor.

El microcontrolador ATmega328P deberá leer el valor de un potenciómetro que actúa como referencia y controlar un motor conectado a un segundo potenciómetro en su eje. El sistema ajustará el giro del motor para igualar el valor del segundo potenciómetro al del primero, utilizando un control por modulación por ancho de pulso (PWM) para variar la velocidad de giro del motor. Por último, se realizaron los gráficos de los valores del pwm y de los potenciómetros en tiempo real a través de la librería pyserial.

Algunas limitaciones que se podrán llegar a encontrar será la precisión del motor al controlarse con PWM para alcanzar la igualdad entre ambos potenciómetros, la velocidad máxima de los ajustes del motor se verá limitada por las características propias del hardware y el control del PWM.

1) Codigo y simulaciones: El codigo en lenguaje C de bajo nivel, implementa la comunicacion UART, el uso de dos Conversores Analogo Digitales (ADC) para la lectura de datos de los potenciometros, asi como un pin para el PWM del motor. Tambien fue necesario utilizar un Puente H, que ayudara al motor DC del kit Fischertechnik a alcanzar un funcionamiento optimo. A continuacion, se observa una imagen de la simulacion utilizada para modelar el circuito que luego fue llevado a un montaje fisico.

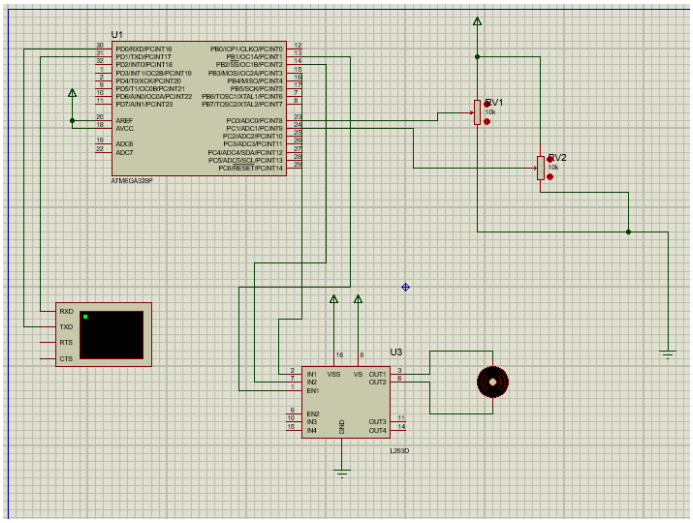

Fig. 6. Circuito simulado del Problema C.

Al utilizar el simulador Proteus, este no posee motores exactamente con el funcionamiento de el del kit Fischertechnik, se ve como al cambiar los valores de los potenciometros, el motor dc cambia de sentido. Esto se puede apreciar en la simulacion adjunta este informe, al igual que el codigo definitivo de funcionamiento del circuito.

2) Montaje fisico: Para realizar el montaje fisico de esta parte, fue necesario el motor de kit Fischertechnik, un integrado puente H, dos potenciometros de igual valor asi los ADC podran realizar las medidas de forma correcta, y el microcontrolador ATmega328P.

Como se ve en la ”Figura 7”, los potenciometros fueron conectados directamete al voltaje y a tierra por sus extremos, y la pata del medio (que sera la que presente cambio en la resistencia), fue conectada directamente al pin A0 y A1 del Arduino, donde se leera el valor de los mismos. El motor DC, fue conectado a una de las salidas del puente H, mientras que el PWM del pin PB1, y los pines PB2 y PB3, que seran los encargados de la direccion del motor, seran conectados a IN1, IN2 y EN, del integrado. Ademas, este integrado requiere una alimentacion de 9V que sera lo necesario que consuma el motor para funcionar de forma correcta.

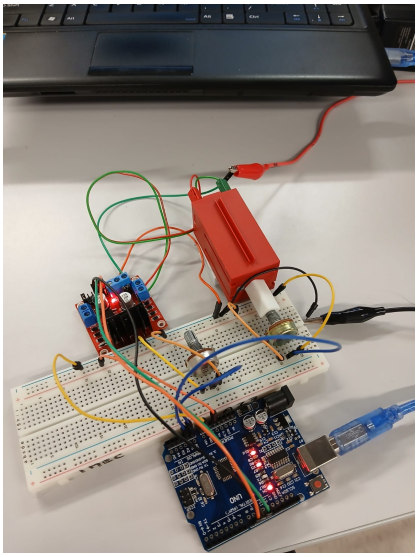

Fig. 7. Montaje fisico del Problema C.

D. Problema D

Para realizar este problema se utilizarán una fotoresistencia, el microcontrolador ATmega328P, un Servomotor y algunos Leds de luz blanca, además de algunas resistencias necesarias para realizar el montaje. El objetivo planteado para este problema, como se encuentra en los apartados anteriores, será la correcta lectura de un color con la fotoresistencia, lo cual hará que el servomotor se mueva para apuntar al color asociado a esos valores. El servomotor será accionado mediante un PWM configurado para recibir estos valores leídos por un ADC y convertirlos a voltaje.

Los colores que fueron elegidos de la hoja de colores dada por el profesor fueron el verde, el amarillo y el rojo, siguiendo una logica de en caso de ser necesario se podria accionar la fotoresistencia a base de Leds de estos colores.

Para la realizar una solución fue necesario realizar un programa en el software Microchip Studio, utilizado para programar en lenguaje c, en esta circunstancia. Luego, se realizaron diferentes pruebas en el sistema de simulación Proteus, llegando al diseño de un circuito funcional como se ve en la “Figura 12”. Por último, este circuito fue llevado a una instancia de funcionamiento físico como se podrá apreciar en la “Figura 9” y en el video adjunto a este informe en la plataforma GitHub.

1) Codigo y simulaciones: El codigo realizado que cumplira la consigna se encontrara en Anexos, asi como en el repositorio de GitHub, al igual que las simulaciones realizadas en Proteus. Aun asi podemos observar un pequeño extracto de la funcion principal del codigo a continuacion y de la simulacion en la ”Figura 12”.

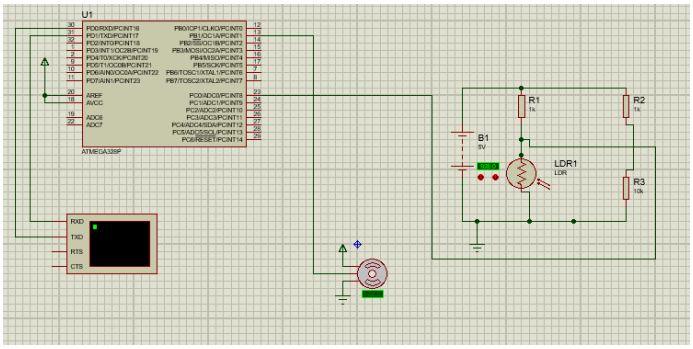

Fig. 8. Circuito de la simulacion del funcionamiento del servomotor y la fotoresistencia en Proteus.

```
    int main(void)
{
        //Inicializamos las
 configuraciones principales
        setupUART(9600);
        setupADC();
        setupPWM();

        //Declaramos algunas variables
 utiles
         uint16_t valor_leido;
        uint16_t valor_pwm;
         char buffer[10];

  while(1)
    {

        valor_leido = readADC(LDR);
 //Se lee el valor de la LDR

        //Un mensaje para comprobar la
 correcta lectura
        UART_sendString("Inicio de
 lectura");
        UART_sendChar('\r');
        UART_sendChar('\n');
        _delay_ms(2000);

        //A continuacion, un if que
 comparara el valor obtenido de
 la LDR y lo compara con un
 color determinado,
        //convirtiendo el valor del
 ADC a un voltaje para el PWM
 que movera el Servo Motor
```

```
 apuntando un color especifico.
    if ((valor_leido>= 800) &&
    (valor_leido<= 899))
        {
          UART_sendString("Color: Rojo");
          UART_sendChar('\r');
          UART_sendChar('\n');
          _delay_ms(2000);

          valor_pwm = ((valor_leido *
   2000) / 1023 );
          setPWM(valor_pwm);
          UART_sendString("Valor pwm: ");
          intToStr(valor_pwm, buffer);
          UART_sendString(buffer);
          UART_sendChar('\r');
          UART_sendChar('\n');

          intToStr(valor_leido, buffer);
// Convierte el valor del ADC a un string
          UART_sendString("Valor leido:");
          UART_sendString(buffer);

          UART_sendChar('\r');
          UART_sendChar('\n');
          _delay_ms(2000);
          }else if ((valor_leido>= 900)
   && (valor_leido<=980))
          {...}
}
```

2) Montaje Fisico: Para el montaje físico, fue necesario emplear un puente de Wheatstone para que la señal recibida por el LDR sea leída de forma correcta por el Conversor Análogo Digital del microcontrolador. Los cálculos de resistencia se encuentran adjuntos a este informe. Los pines que fueron utilizados del microcontrolador fueron: el pin A0, del puerto análogo donde se ubicará el ADC de entrada de datos, el pin PB1 para la salida del PWM hacia el servomotor, los pines de comunicación serial PD0 y PD1, y el pin de AREF para la configuración del ADC. Para realizar el puente de Wheatstone se utilizaron 2 resisencias de 10k Ω, la fotrresitencia y una resistencia de un 1M Ω, este valor resultara mas alto del calculado ya que sera la forma de que se vea la señal en ADC.

VI. Resultados

A. Problema A

Durante la implementación del sistema de control del plotter, se lograron obtener los resultados esperados. A través del menú interactivo desplegado por comunicación serial, el usuario pudo seleccionar entre las diferentes figuras predefinidas: triángulo, círculo, cruz, perro y conejo.

El sistema mostró un comportamiento preciso en la ejecución de las secuencias de movimiento del plotter, manteniendo una alta precisión en la representación de las figuras, especialmente en las líneas rectas. Sin embargo, algunas limitaciones fueron identificadas al momento de trazar curvas, ya que el uso de diagonales en lugar de trayectorias circulares perfectas dio lugar a figuras con curvas aproximadas. A pesar de esto, el objetivo de la práctica se cumplió satisfactoriamente, permitiendo el dibujo automatizado de las figuras seleccionadas.

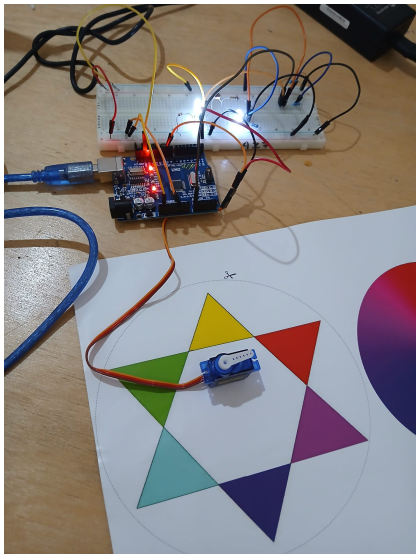

Fig. 9. Circuito fisico del funcionamiento de la LDR y el servomotor.

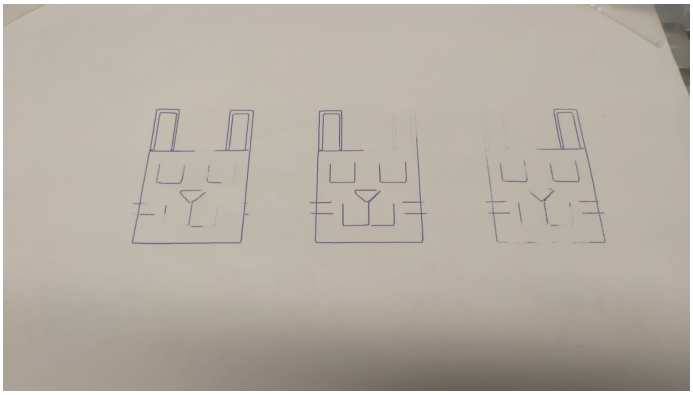

Fig. 10. Ejemplo de dibujo realizado en plotter.

B. Problema B

Para realizar el montaje físico , nos encontramos con ciertas dificultades al momento de la implementación de la amplificación operacional. Esto no impidió que no se entendiera el concepto de cómo implementar el circuito , teniendo en cuenta que el código está realizado con una cierta inexactitud dado que no pudimos obtener los valores de temperatura para realizar la ecuación que nos daría el cálculo de la temperatura. Pero el código cumple con la lógica esperada para el circuito. También destacar que fue probado a través de una simulación y cumpliendo el funcionamiento.

C. Problema C

Al realizar el funcionamiento del montaje fisico, se encongraron ciertas dificultados con los valores del PWM, ya que un PWM de valor 255 seria demasiado y en varias ocasiones, al mover el eje del motor, daño el potenciometro, sacandolo de la Protoboard. Fue necesario realizar distintas pruebas y ajustes para alcanzar un valor optimo de 165 para que se moviera a una velocidad baja sin dañar nada.

Como se observa en el video adjunto al informe, este montaje resultó funcional y util para poder obtener los valores y graficarlos a traves de la libreria Pyserial. Estos graficos seran adjuntados al informe en Anexos. LAs escalas utilizadas para modelar los valores fueron de 1023 para los valores de los potenciometros y de 255 para los valores de PWM.

D. Problema D

Realizando un breve analisis de esta parte se puede apreciar que se cumple con el objetivo de mover el Servomotor a partir de los valores leidos de la fotoresistencia, utilizando un PWM para una mayor eficacia.

En cuanto al montaje fisico, se pueden ver a continuacion los calculos realizados para el puente de wheatstone utilizado. La resistencia que se coloco junto a la LDR fue de 1MΩ, ya que sera mayor a la maxima resistencia alcanzada por la LDR en la maxima oscuridad. Se encuentra un grafico de acondicionamiento de la LDR en la ”Figura 14”.

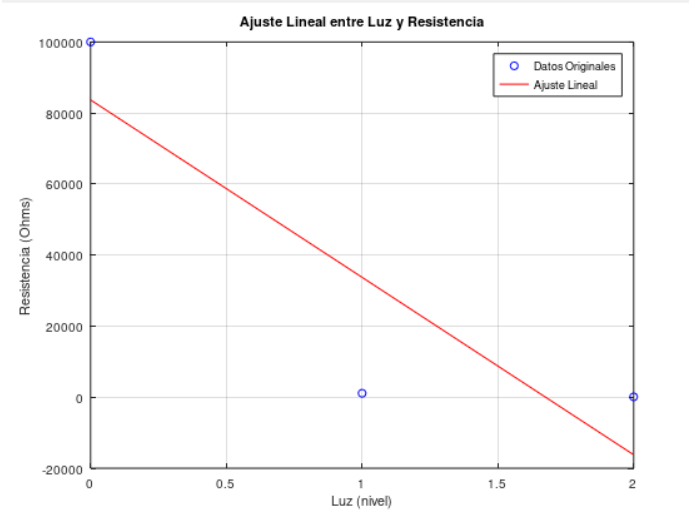

Fig. 11. Grafico de acondicionamiento de la fotoresistencia.

Para calcular el puente de wheatstone se realizaron dos divisores de voltaje entre una reistencia de 10k Ω y la LDR, y entre la otra resistencia de 10kΩ y la resitencia a la que le adjudicaremos un valor mas alto del minimo hallado. Estos calculos se encuentran adjuntados al informe.

Por ultimo, comentar que el montaje fue capaz de leer los valores de la hoja de colores y mover el servomotor segun el color de referencia apuntado.

VII. Conclusiones

En el control del motor DC, el uso de dos potenciometros permitió ajustar la velocidad y dirección de manera efectiva, mientras que la visualización en tiempo real de los datos a través de la biblioteca PySerial facilitó la comprensión del comportamiento del sistema. La implementación del circuito de cambio de color, mediante la fotoresistencia y el servomotor controlado por PWM, evidenció la capacidad del sistema para adaptarse a las variaciones en la luz ambiental, logrando un posicionamiento preciso del servomotor.

Por otro lado, la planta simuladora de cambio de temperatura, basada en el sensor PT100, aunque no fue posible su montaje fisico, fue posible realizar un codigo que simula el funcionamient de forma bastante adecuada. Por ultimmo, el desarrollo del plotter permitió visualizar gráficamente los resultados.

References

[1] Repositorio de GitHub. https://github.com/BelenBoado/Tecnolog-as-de-Microprocesamientos/tree/inicializacion/Laboratorio%202

[2] Informacion sobre microcontrolador ATmega328P. https://www.incb.com.mx/index.php/articulos/78-microcontroladores-y-dsps/2546-conociendo-el-microcontrolador-nucleo-core-atmega328p-de-arduino-uno-mic019s

[3] Informacion sobre microcontrolador ATmega328P. https://es.wikipedia.org/wiki/Atmega328

[4] Informacion sobre el lenguaje de programacion C. https://openwebinars.net/blog/que-es-c/

[5] Informacion sobre la PT100. https://www.fujielectric.fr/es/tecnologias/sensor-de-temperatura-pt100/#:~:text=La%20PT100%20toma%20su%20nombre,ohmios%20a%200%C2%B0C.

[6] Informacion sobre Plotter. https://digipressystem.com/que-es-un-plotter-utilidades-tipos-diferencias/#:~:text=Un%20plotter%20de%20impresi%C3%B3n%2C%20tambi%C3%A9n,y%20rotulaci%C3%B3n%20de%20todo%20tipo.

[7] Informacion sobre Servomotores. https://www.a-m-c.com/es/servomotor/

[8] Informacion sobre integrado Puente H. https://laelectronica.com.gt/extras/que-es-y-como-funciona-un-l293d?srsltid=AfmBOoq2ctxZl5rDOHV0MlVaxmWI850rPb-1-R5HcrMKx10at9RnwPZR

VIII. Anexos

```
Script del grafico de acondicionamiento
de la LDR:
r = [100e3 1.1e3 0.100e3];
luz = [0 1 2];

p = polyfit(luz, r, 1);

r_ajuste = polyval(p, luz);

plot(luz, r, 'ob', 'markersize', 4,
'DisplayName', 'Datos Originales');
hold on;
plot(luz, r_ajuste, '-r',
'DisplayName', 'Ajuste Lineal');
xlabel('Luz (nivel)');
ylabel('Resistencia (Ohms)');
title('Ajuste Lineal entre Luz y
Resistencia');
legend('show');
grid on;
```

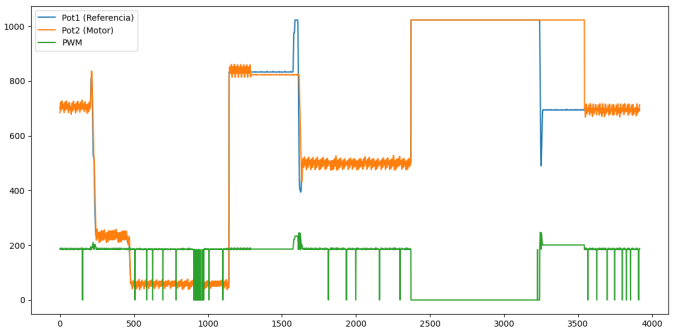

Fig. 12. Primer grafico obtenido de los valores medidos por los potenciometros y el PWM.

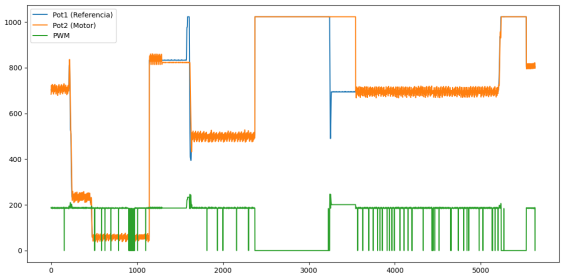

Fig. 13. Segundo grafico obtenido de los valores medidos por los potenciometros y el PWM.

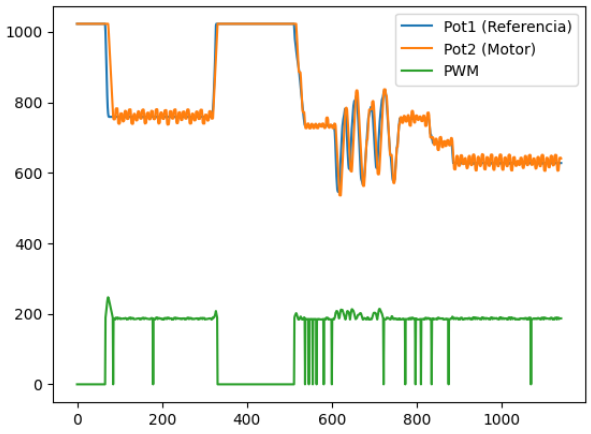

Fig. 14. Tercer grafico obtenido de los valores medidos por los potenciometros y el PWM.