2do laboratorio

Rodrigo Karlen
UTEC
Ingeniería Mecatrónica
Fray bentos, Uruguay
rodrigo.karlen@estudiantes.utec.edu.uy

Francisco Mayol
UTEC
Ingeniería Mecatrónica
Fray bentos, Uruguay
francisco.mayol@estudiantes.utec.edu.uy

Luis Nunez Da Rosa
UTEC
Ingeniería Mecatrónica
Fray bentos, Uruguay
luis.nunezdarosa@estudiantes.utec.edu.uy

I. Resumen

En este informe se documenta el desarrollo de un sistema de control basado en el microcontrolador ATmega328P, con el objetivo de abordar una serie de problemas relacionados con el control de movimientos, temperatura y motores, y la detección de colores. El laboratorio se dividió en diferentes secciones, cada una enfocada en la implementación y análisis de sistemas mecatrónicos específicos.

En la primera parte, se exploró el funcionamiento de un plotter, controlado mediante relés electromecánicos. Se logró implementar movimientos precisos en diferentes direcciones, ajustando tiempos de conmutación y explorando movimientos diagonales y angulares. Esto permitió dibujar figuras geométricas predefinidas con éxito, controlando también la activación de la punta del plotter.

La segunda parte del laboratorio abordó cuatro problemas clave: el control del plotter mediante un menú interactivo y figuras programadas, la regulación automática de temperatura utilizando un sensor PT100 y actuadores (calefactor y ventilador), el control de un motor mediante la lectura de potenciómetros y la aplicación de PWM, y finalmente, un sistema de detección de colores mediante fotocelda, que ajustaba la posición de un servomotor en función del color detectado.

A lo largo del laboratorio se realizaron ajustes y pruebas para asegurar la precisión y funcionamiento óptimo de cada sistema implementado, cumpliendo con los objetivos planteados.

II. Introducción

En la presente práctica se explorará el funcionamiento de un plotter monocromático controlado por un microcontrolador ATmega328P. El objetivo es familiarizarse con el control de los movimientos del plotter a través de relés electromecánicos, permitiendo realizar trazos y dibujos predefinidos. Además, se trabajará con la comunicación serial entre el microcontrolador y el PLC del plotter, garantizando la precisión en el movimiento de los motores y el accionamiento de la válvula neumática para el trazado.

La práctica también contempla la programación y control del ATmega328P para resolver problemas adicionales, utilizando sensores y actuadores en sistemas automatizados. Esto incluye la regulación de temperatura, el control de motores a través de potenciómetros, y la detección de colores mediante una fotocelda, con el fin de simular escenarios reales de automatización.

III. Objetivos

Objetivo Principal

Implementar un sistema de control utilizando el ATmega328P que permita resolver distintos problemas mecatrónicos relacionados con el movimiento, temperatura, motores y sensores ópticos.

Objetivos Específicos

- Comprender el funcionamiento del plotter utilizado en el laboratorio, enfocándose en su control mediante el microcontrolador ATmega328P.
- Solucionar cuatro problemas relacionados con el control de movimiento, regulación de temperatura, control de velocidad de motores y detección de colores, utilizando el microcontrolador ATmega328P y comunicación serial.
- Desarrollar un protocolo de comunicación eficiente entre los sensores y actuadores conectados al ATmega328P, asegurando respuestas rápidas y coherentes del sistema.

IV. Materiales

En esta práctica utilizaremos los siguientes materiales, necesarios para la implementación y control de los distintos sistemas a realizar:

- Microcontrolador ATmega328P
- Controlador Lógico Programable (PLC)
- Plotter del Laboratorio de Mecatrónica
- Sensores de movimiento
- Relés electromecánicos
- Pulsadores
- Válvula neumática
- Motor paso a paso
- Cables para conexiones eléctricas
- Fuente de alimentación
- Computadora con software de programación
- Potenciómetros
- Sensor de temperatura
- Fotocelda
- Servomotor
- Controlador de motores L298N

V. Marco Teórico

ATmega328P

El microcontrolador ATmega328P es un chip de 8 bits basado en la arquitectura AVR, ampliamente utilizado en sistemas embebidos por su versatilidad y eficiencia. Entre sus principales características destacan sus 32 KB de memoria flash, 2 KB de SRAM, y 1 KB de EEPROM, lo que le permite almacenar y ejecutar programas de tamaño moderado. Además, cuenta con 23 pines de entrada/salida digital, 6 de los cuales pueden funcionar como salidas PWM, y 6 canales ADC de 10 bits. También soporta comunicación en serie mediante UART, SPI y I2C. Se adjunta en la Figura 1 una imagen del microcontrolador.

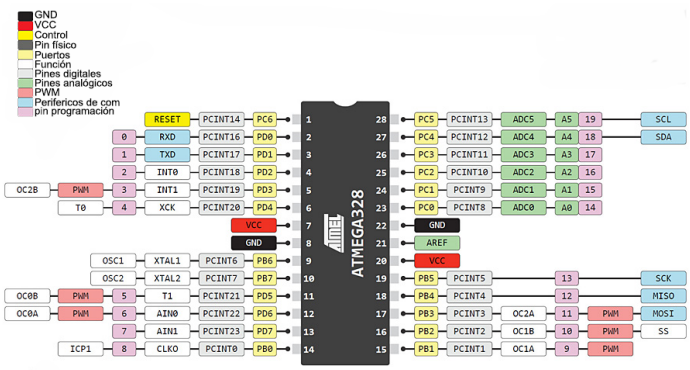


Fig. 1. Microcontrolador Atmega328P

Controladores Lógicos Programables (PLC)

Los Controladores Lógicos Programables (PLC) son dispositivos diseñados para la automatización de procesos industriales. Funcionan monitoreando entradas de diversos sensores o dispositivos de control, procesando la información mediante un programa previamente configurado y actuando sobre salidas que controlan equipos como motores, relés o válvulas.

Una de sus principales ventajas es la robustez frente a condiciones adversas, lo que los hace ideales para entornos industriales. Además, su capacidad de reprogramación permite ajustar su funcionamiento a diferentes tareas sin necesidad de cambiar el hardware.

Los PLCs operan en tiempo real, lo que garantiza que las acciones de control se ejecuten de manera precisa y en el momento adecuado. Esta tecnología es ampliamente utilizada en la automatización de fábricas, plantas de procesamiento, y cualquier sistema que requiera un control secuencial y confiable de múltiples componentes.

Sensor de Temperatura PT100

El sensor de temperatura PT100 es un tipo de termorresistencia o RTD (del inglés Resistance Temperature Detector), ampliamente utilizado en la medición precisa de temperatura en aplicaciones industriales. El principio de funcionamiento del PT100 se basa en la variación de la resistencia eléctrica del platino en función de la temperatura: a medida que la temperatura aumenta, también lo hace la resistencia del elemento sensor.

El nombre PT100 proviene de la composición del sensor (platino, Pt) y el valor de su resistencia a 0°C, que es de 100 ohmios. Este tipo de sensor es muy apreciado por su alta precisión y estabilidad a lo largo de un amplio rango de temperaturas, que típicamente varía entre -200°C y 850°C.

Debido a su fiabilidad y precisión, el PT100 se utiliza en una gran variedad de aplicaciones industriales, incluyendo el control de procesos, la supervisión de sistemas de calefacción y la refrigeración. Su respuesta lineal a los cambios de temperatura lo convierte en una opción preferida para entornos donde es fundamental un control exacto de la temperatura.

Se adjunta en la Figura 2 una imagen del sensor mencionado.

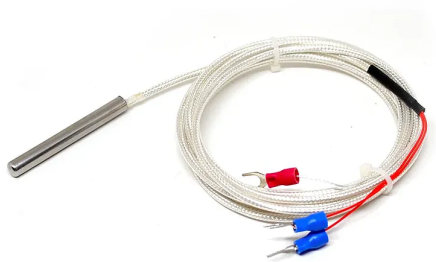


Fig. 2. Sensor de Temperatura PT100

Controlador de motores L298N

En la implementación, fue imperativo el uso del controlador de motores L298N, el mismo es un dispositivo que permite manejar motores de corriente continua (DC) y motores paso a paso. Funciona recibiendo señales de control desde un microcontrolador, en este caso del Atmega328p. El L298N utiliza estas señales para regular la dirección y la velocidad de los motores conectados. El L298N puede controlar dos motores a la vez, gracias a sus dos puentes H integrados, que permiten invertir la polaridad de la corriente y así cambiar la dirección de giro. Además, cuenta con pines de entrada para habilitar y deshabilitar los motores, así como para ajustar la velocidad mediante la modulación por ancho de pulso (PWM). Se adjunta en la Figura 3 una imagen del controlador mencionado.

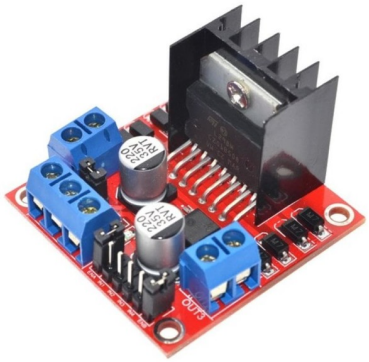


Fig. 3. Controlador de motores L298N

Plotter

Un plotter es un dispositivo de impresión especializado en la generación de gráficos vectoriales mediante el trazado de líneas continuas, lo que lo diferencia de las impresoras convencionales, que trabajan a base de puntos. Utiliza un sistema de plumas o cabezales controlados por motores que se desplazan a lo largo de los ejes X y Y para dibujar sobre superficies como papel o vinilo. Debido a su capacidad para producir gráficos de alta precisión y detalles finos, los plotters son comúnmente empleados en áreas como el diseño asistido por computadora (CAD), arquitectura, ingeniería y la fabricación de mapas y planos técnicos. Además, son ideales para trabajos que requieren representaciones exactas a gran escala, como planos arquitectónicos o diagramas eléctricos.

Se adjunta en la Figura 4 una imagen del plotter del Laboratorio de Mecatrónica.

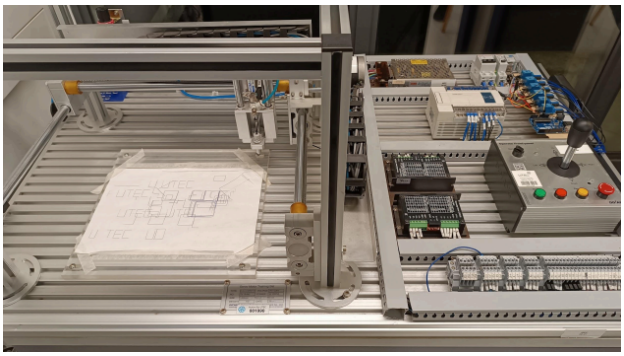


Fig. 4. Plotter del Laboratorio de Mecatrónica

Servomotor SG90

El servomotor SG90 es un motor de tipo servo compacto y de bajo costo, ampliamente utilizado en aplicaciones de control de movimiento. Este dispositivo es capaz de posicionar su eje en un ángulo específico, generalmente dentro de un rango de 0 a 180 grados, con alta precisión. Está compuesto por un motor de corriente continua, un sistema de engranajes y un circuito de control que regula la posición del eje a través de una señal de modulación por ancho de pulso (PWM). Debido a su tamaño reducido y su capacidad de respuesta rápida, el servomotor SG90 es ideal para proyectos de robótica, automatización y dispositivos que requieren un control preciso de ángulos, como brazos robóticos, sistemas de dirección y mecanismos de control de superficies en vehículos aéreos. Se adjunta en la Figura 5 una imagen del Servomotor SG90.

Fig. 5. Servomotor SG90

Amplificador multiplicador no inversor

El amplificador no inversor es un tipo de amplificador operacional donde la señal de entrada se aplica a la entrada no inversora del Op-Amp (+). Este tipo de configuración ofrece una amplificación sin invertir la señal de entrada, es decir, la señal de salida está en fase con la señal de entrada.

El amplificador multiplicador no inversor sigue la ecuación:

$$V_{out} = V_1 \left(1 + \frac{R_f}{R_1}\right)$$

Aquí, V_1 es la señal de entrada, R_f es la resistencia de realimentación y R_1 es la resistencia conectada a tierra. Este circuito permite multiplicar la señal de entrada por un factor determinado por la relación entre R_f y R_1. Se utiliza en aplicaciones donde se necesita aumentar la magnitud de una señal sin cambiar su polaridad, como en sistemas de procesamiento de señales y filtros activos.

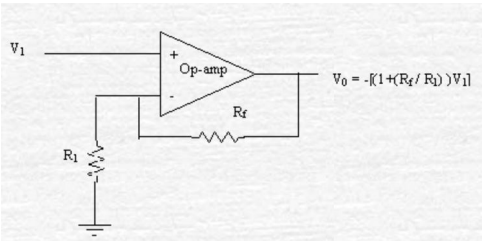


Fig. 6. Configuración amplificador multiplicador no inversor

Amplificador restador

El amplificador restador es un circuito basado en un amplificador operacional (Op-Amp) que permite obtener la diferencia entre dos señales de entrada, V_1 y V_2, amplificadas por resistencias R. La salida del circuito, V_{out}, es proporcional

a la diferencia de potencial entre las dos señales aplicadas a las entradas del Op-Amp. Este tipo de amplificador tiene la ventaja de restar directamente las señales de entrada, con la fórmula general de salida:

$$V_{out} = \frac{R_f}{R}(V_2 - V_1)$$

donde R_f es la resistencia de realimentación del amplificador.

Los amplificadores restadores se utilizan comúnmente en sistemas donde es necesario extraer la señal diferencial entre dos puntos, como en las aplicaciones de instrumentación, para eliminar ruido o componentes comunes a ambas señales (señal de modo común).

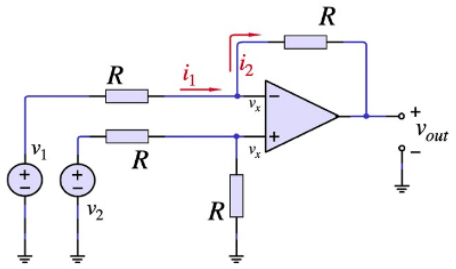


Fig. 7. Configuración amplificador restador

VI. Procedimiento

En esta sección se detallará el procedimiento a seguir para completar con éxito la práctica. La actividad se dividirá en dos partes principales: la primera estará enfocada en el análisis y comprensión del funcionamiento del plotter monocromático, el cual es controlado mediante un PLC y permite la ejecución de movimientos precisos en distintos ejes. La segunda parte abordará cuatro problemas específicos: la implementación de un sistema de control de temperatura, el control de un motor con potenciómetros y PWM, la detección de colores con fotocelda y servomotor, y la programación del plotter para dibujar figuras predefinidas con opciones de personalización a través de un menú interactivo.

Parte 1: Funcionamiento del Plotter

Descripción del Plotter

En el caso particular del Laboratorio de Mecatrónica, se cuenta con un plotter monocromático el cual es operado por un Controlador Lógico Programable (PLC) que será utilizado para llevar a cabo la Parte A de este laboratorio.

Funcionamiento del mismo

Los sensores y actuadores del plotter se encuentran conectados al PLC, por lo que se dispone para realizar la práctica los accionamientos de los motores (arriba, abajo, izquierda y derecha) y el accionamiento de la válvula neumática para habilitar el trazado. Cada vez que el PLC es reiniciado o se quita la alimentación, al encender este volverá a una posición inicial, siendo esta en el extremo superior derecho. También está configurado un pulsador de RESET para devolver el puntero del plotter a la posición antes mencionada.

IMPORTANTE: Se tiene definido un rango de dibujo, por lo que se debe respetar el mismo. El PLC posee la capacidad de detener el dibujo si se llega a los límites de la hoja.

Parada de emergencia: El plotter posee un pulsador de emergencia, E-STOP, el cual tiene la particularidad de detener toda la secuencia de movimientos dejando al plotter detenido en el último punto antes de ser llamada la emergencia.

Conexiones del microcontrolador con el plotter

Se hará uso de relés electromecánicos para enviar señales al PLC desde el microcontrolador. En la Tabla 1 se describen los pines y sus conexiones correspondientes hacia el plotter, Es importante mantener esta organización, ya que el plotter del laboratorio de mecatrónica ya tiene el Arduino integrado y conectado de esta forma.

TABLE I
CONEXIONES DEL MICROCONTROLADOR CON EL PLOTTER

Pin Digital	Conexión
D2	Bajar solenoide X0
D3	Subir solenoide X1
D4	Movimiento hacia abajo X5
D5	Movimiento hacia arriba X6
D6	Movimiento hacia la izquierda X7
D7	Movimiento hacia la derecha X10

En la Fig. 6 se aprecia el control manual del plotter.

Fig. 8. Control del plotter

Parte 2: Problemáticas

La Parte 2 se dividirá en cuatro secciones principales: la primera, denominada Problema A, estará enfocada en la implementación del sistema de control para el plotter, donde se programarán las diferentes figuras predefinidas y la comunicación mediante serial. La segunda sección, Problema B, se centrará en el control de la temperatura utilizando un sensor PT100, incluyendo la regulación automática mediante un calefactor y ventilador. La tercera sección, Problema C, abordará el control de un motor a través de la lectura de potenciómetros, ajustando la velocidad y dirección mediante PWM. Finalmente, la cuarta sección, Problema D, tratará sobre un sistema de selección de colores utilizando una fotocelda y un servomotor, con

la identificación de colores y su correspondiente control de posición.

Problema A

Descripción General

El objetivo de esta sección es programar un microcontrolador ATmega328P en lenguaje C para controlar un plotter de dibujo mediante comunicación serial. El sistema permitirá al usuario seleccionar y dibujar diversas figuras. Las funcionalidades principales que debe tener el sistema son:

- Mostrar un menú interactivo a través del puerto serial.
- Dibujar las figuras predefinidas: triángulo, círculo y cruz.
- Dibujar las figuras asignadas: perro y murciélago.
- Ejecutar las secuencias de dibujo considerando los tiempos de conmutación de los relés, garantizando precisión en el movimiento del plotter.
- Permitir la edición personalizada de los dibujos por parte del usuario.

Limitaciones

El sistema debe cumplir con las siguientes limitaciones:

- La cantidad de figuras disponibles está limitada a las opciones predeterminadas (triángulo, círculo, cruz) y las figuras asignadas (perro y murciélago).
- La comunicación con el plotter dependerá de la velocidad del puerto serial y la precisión en la conmutación de los relés.
- El tiempo de respuesta puede verse afectado por la complejidad de la figura seleccionada y los límites mecánicos del plotter.

Problema B:

El objetivo de esta sección es implementar un sistema de control automático de temperatura utilizando un microcontrolador ATmega328P y un sensor PT100. El sistema deberá realizar lecturas periódicas de temperatura y controlar, de forma automática, un calefactor y un ventilador para mantener la temperatura ambiente dentro de los rangos definidos.

Descripción del Sistema

El sistema tomará una medición de temperatura cada 5 segundos utilizando el sensor PT100. Según los valores obtenidos, se activarán los dispositivos de control de temperatura de acuerdo con los siguientes criterios:

- **0 a 22°C:** El calefactor se encenderá para aumentar la temperatura hasta alcanzar el rango adecuado.
- **23 a 30°C:** En este rango, tanto el calefactor como el ventilador permanecerán apagados, ya que la temperatura se considera óptima.
- **31 a 40°C:** El ventilador se encenderá a baja velocidad para reducir gradualmente la temperatura.
- **41 a 50°C:** El ventilador operará a velocidad media para acelerar la reducción de temperatura.
- **Más de 51°C:** El ventilador se activará a su máxima velocidad para evitar un sobrecalentamiento.

El sistema enviará información a través de la comunicación serial, indicando la temperatura medida y las acciones realizadas (activación del calefactor o del ventilador).

Ajustes Adicionales

El sistema permite al usuario modificar el ”punto medio” de temperatura mediante un menú interactivo accesible por comunicación serial. Esta función recalibra automáticamente los rangos de temperatura superior e inferior, garantizando un control más flexible y ajustado a las necesidades del entorno sin requerir modificaciones en el código.

Visualización de Resultados

Para analizar el comportamiento del sistema, se registrarán y graficarán las temperaturas medidas a lo largo del tiempo utilizando herramientas como Python o MATLAB. La gráfica deberá incluir:

- La evolución de la temperatura.
- Las acciones del sistema (encendido/apagado del calefactor y la velocidad del ventilador).
- El rango de temperatura óptimo definido según el ”punto medio” establecido por el usuario.

Esta representación gráfica permitirá verificar la estabilidad del sistema y su capacidad para mantener la temperatura dentro de los rangos deseados.

Problema C:

El microcontrolador ATmega328P se encargará de leer el valor de un potenciómetro que actúa como referencia y controlará 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.

Descripción del Sistema

El sistema funcionará midiendo los valores de dos potenciómetros:

- El primer potenciómetro define el valor de referencia.
- El segundo potenciómetro, acoplado al eje del motor, mide la posición actual del motor.

El microcontrolador ATmega328P ajustará la velocidad y el sentido de giro del motor mediante PWM, con el objetivo de igualar el valor del segundo potenciómetro al del primero.

El sistema enviará, a través de la comunicación serial, los siguientes parámetros:

- Valor del potenciómetro de referencia.
- Valor del potenciómetro acoplado al motor.
- Valor del PWM aplicado al motor.
- Sentido de giro del motor (horario o antihorario).

Funcionamiento Detallado

1) El primer potenciómetro define el valor de referencia que se quiere alcanzar.
2) El segundo potenciómetro, acoplado al eje del motor, mide la posición actual del motor.
3) El microcontrolador ajusta la velocidad y el sentido de giro del motor utilizando la señal PWM hasta que el valor de ambos potenciómetros se iguale.
4) Durante el proceso, el sistema muestra por puerto serial los valores de ambos potenciómetros, el valor de PWM y el sentido de giro del motor.

Visualización de Resultados

Se deben graficar los valores del potenciómetro de referencia, del potenciómetro en el eje del motor y del PWM aplicado al motor. Las gráficas serán generadas utilizando Python o MATLAB para facilitar el análisis del comportamiento del sistema.

Limitaciones

La precisión del sistema para igualar los valores del potenciómetro de referencia y el acoplado al motor está directamente influenciada por la resolución de los potenciómetros y la calidad del control PWM. Si la diferencia inicial entre los valores de ambos potenciómetros es significativa, el sistema puede tardar más en ajustarlos. Además, la velocidad máxima a la que el motor puede hacer estos ajustes está limitada tanto por las características del hardware disponible como por el control PWM implementado.

Problema D:

El sistema consiste en un selector de colores utilizando un microcontrolador ATmega328P, una fotocelda y un servomotor. El objetivo es detectar el color presente en una hoja de referencia mediante la fotocelda y, en función del color identificado, posicionar el servomotor en un ángulo determinado.

Descripción del Sistema

El sistema funcionará de la siguiente manera:

- El microcontrolador ATmega328P leerá los valores de la fotocelda, que actuará como sensor de luz, detectando el color de la hoja de referencia.
- Según el color identificado, el servomotor se moverá al ángulo correspondiente previamente definido.
- Este sistema estará configurado para reconocer una cantidad limitada de colores, correspondientes a los presentes en la hoja de referencia. A cada color se le asignará un valor predefinido de conversión analógica a digital (ADC) para garantizar una identificación precisa.

El sistema enviará, a través de la comunicación serial, los siguientes parámetros:

- Valor de la fotocelda.
- Color detectado.
- Diferencia entre valor establecido y valor de lectura.

VII. Resultados

En esta sección se presentan los resultados obtenidos durante la práctica, siguiendo el procedimiento descrito anteriormente. Los resultados están organizados de acuerdo con las distintas fases de la práctica, para facilitar su análisis y comprensión.

Repositorio GitHub

En este apartado se presenta el enlace al repositorio en GitHub, donde se encuentran todos los códigos y las evidencias necesarias solicitadas para la práctica. A lo largo de la exposición de los resultados, se hará referencia a diferentes carpetas y archivos dentro del repositorio. Este contiene múltiples versiones de los códigos conforme se fue avanzando en el proyecto, además de incluir imágenes y vídeos que demuestran detalladamente, el funcionamiento de las tareas solicitadas.

En el caso específico de este laboratorio, todos los archivos mencionados se encuentran ubicados dentro de la carpeta *Laboratorio 2*, que a su vez está dentro de la carpeta *Laboratorios*. Allí, se encuentran organizadas las carpetas correspondientes que se irán mencionando a medida que se presenten los resultados, permitiendo un acceso claro y estructurado a los códigos y evidencias generadas durante la práctica.

Enlace: ***Link a Repositorio GitHub***.

Parte 1: Funcionamiento del Plotter

Al inicio de la práctica, el enfoque principal fue entender el funcionamiento del plotter y su control mediante relés. Se confirmó que el desplazamiento del plotter en las cuatro direcciones principales (arriba, abajo, derecha e izquierda) dependía del tiempo durante el cual se mantenía activado cada uno de los motores.

Además, se consideró la necesidad de incluir un tiempo de espera entre las instrucciones para evitar sobrecargar los relés y asegurar su correcto funcionamiento. Junto con el control de los movimientos direccionales, también fue necesario accionar la válvula neumática para controlar el descenso y ascenso de la punta del plotter sobre el papel, siguiendo la distribución de pines establecida en el procedimiento.

Se exploró, además, la posibilidad de realizar trayectorias diagonales, lo cual era fundamental para generar las figuras propuestas. Tras analizar el sistema, se concluyó que para trazar una diagonal era necesario conmutar rápidamente entre un movimiento vertical y uno horizontal. Por ejemplo, al accionar el motor hacia la izquierda y luego, de forma rápida, el motor hacia arriba, se consiguió una diagonal ascendente hacia la izquierda.

El principal desafío fue encontrar el tiempo adecuado entre cada conmutación de los relés para evitar errores en los movimientos y lograr con éxito los desplazamientos diagonales. Tras varias pruebas, se determinó que el tiempo ideal era de 70 milisegundos en una dirección, seguido de 70 milisegundos en la otra. Esto permitió un control preciso y fluido de las cuatro diagonales posibles (arriba-derecha, arriba-izquierda, abajo-derecha, abajo-izquierda). Es importante destacar que

las líneas resultantes formaban siempre un ángulo de 45 grados respecto a los ejes horizontal y vertical, lo que limitaba el movimiento del plotter a esas direcciones.

Sin embargo, al analizar cómo dibujar un círculo, surgió la necesidad de generar diagonales con ángulos diferentes a los 45 grados para mejorar la precisión del trazo. A través de más pruebas, llegamos a la conclusión de que, variando los tiempos de activación de los motores para los movimientos vertical y horizontal, era posible ajustar el ángulo de inclinación. Ajustando los milisegundos que se mantenían encendidos los motores, logramos obtener desplazamientos en ángulos de 30 y 60 grados respecto a los ejes. Esto nos permitió superar la limitación inicial de las ocho direcciones principales y conseguir un movimiento más flexible, logrando así un trazado de círculos mucho más preciso.

Gracias a este análisis exhaustivo, basado en la comprensión del sistema y en pruebas iterativas, se logró dominar completamente el funcionamiento del plotter, lo que permitió avanzar de manera efectiva en la resolución del Problema A.

Parte 2:

A partir de ahora, se presentarán los resultados obtenidos al realizar los cuatro problemas planteados en el procedimiento. Estos resultados estarán organizados siguiendo el mismo orden en que fueron presentados en la sección de procedimiento, con el fin de mantener una estructura clara y coherente en el análisis de cada uno de los problemas.

Problema A

En esta sección se exponen los resultados obtenidos tras la implementación del código para controlar el plotter y realizar las figuras solicitadas. Primero, se realizaron diversas pruebas para ajustar los tiempos de activación de los motores y estudiar el comportamiento del plotter. Como se explicó en la sección anterior, se consiguió una comprensión completa del funcionamiento del plotter, lo que permitió ejecutar todos los movimientos necesarios. Una vez logrado esto, se avanzó en comprender cómo transformar el control de tiempo en control espacial, es decir, cómo traducir los tiempos de activación de los motores en desplazamientos precisos sobre la hoja.

Durante la implementación del código, se desarrollaron varias funciones esenciales para el correcto control del plotter. En primer lugar, se implementaron funciones específicas para activar y desactivar el solenoide, permitiendo controlar el descenso y ascenso de la punta del plotter sobre la hoja. Además, se crearon las funciones necesarias para la comunicación serial a través del protocolo UART. También se implementaron las funciones para manejar los movimientos en las distintas direcciones (vertical, horizontal y diagonal), ajustando el tiempo de activación de los motores para lograr los desplazamientos deseados. Finalmente, se desarrolló una función clave para la creación de un menú interactivo, accesible a través del puerto serial, que permitía al usuario seleccionar y ejecutar las figuras predefinidas con facilidad.

Mediante el análisis del comportamiento del plotter en movimientos horizontales, verticales y diagonales, se desarrollaron distintas funciones para controlar cada uno de estos movimientos, utilizando parámetros de tiempo en milisegundos. Para los movimientos diagonales, se emplearon bucles for, donde el número de iteraciones controlaba la duración y precisión del desplazamiento. Se concluyó que para mover el plotter 1.1 cm en una dirección horizontal o vertical, era necesario pasar 1000 ms como parámetro, mientras que para lograr un desplazamiento de 1 cm en diagonal, se requería un ciclo de 10 iteraciones. Asimismo, para movimientos diagonales de 30 y 60 grados, un parámetro de 20 iteraciones resultaba en un desplazamiento de 2.8 cm.

Con estos datos, se realizaron bocetos de las figuras sobre papel, midiendo las distancias en centímetros y luego traduciendo esos valores a tiempos e iteraciones en el código. De esta forma, fue posible realizar con éxito las figuras del triángulo, la cruz y el círculo.

Para las figuras más complejas, como el perro y la manzana, fue necesario un mayor grado de precisión. Además de controlar los movimientos del plotter, se tuvo que gestionar cuidadosamente el descenso y ascenso de la punta mediante la activación y desactivación del solenoide en los momentos exactos. El control preciso de este elemento fue crucial para evitar que los trazos se interrumpieran o superpusieran, lo que requería una sincronización meticulosa entre los movimientos y el solenoide.

Se implementó también un menú interactivo accesible mediante la comunicación UART por el puerto serial. Este menú permitía al usuario seleccionar la figura a dibujar introduciendo un número del 1 al 5, con las opciones correspondientes a triángulo, círculo, cruz, perro y manzana. Esto ofrecía al usuario la posibilidad de seleccionar y ejecutar las figuras de manera sencilla, asegurando además que el plotter se desplazara a posiciones iniciales adecuadas para evitar que las figuras se dibujaran sobre los trazos anteriores.

De esta manera, se logró implementar un sistema completo que permite controlar el plotter de manera eficiente y precisa. Cada uno de los componentes del código, desde las funciones para activar el solenoide hasta el menú interactivo, cumplió su objetivo facilitando el dibujo de las figuras solicitadas. A través del ajuste de los tiempos y las pruebas realizadas, se obtuvieron los parámetros necesarios para traducir los comandos en movimientos exactos, lo que permitió la correcta ejecución de todas las figuras, incluidas las más complejas, como el perro y la manzana.

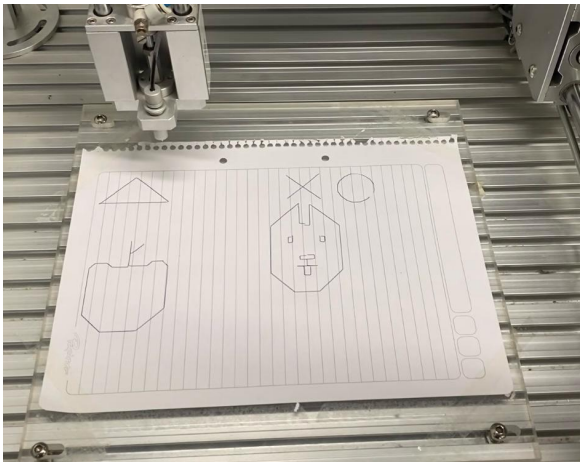


Fig. 9. Figuras dibujadas mediante Plotter

Para una mayor comprensión del desarrollo y ejecución del Problema A, se recomienda revisar el ***código*** generado y observar el montaje y funcionamiento completo del plotter, incluyendo la elaboración de todas las figuras solicitadas, en el video presentado en el repositorio disponible en el siguiente enlace. Ambos archivos están disponibles en el repositorio de GitHub previamente compartido.

Enlace: ***Video Problema A***.

Problema B

En esta instancia de la actividad práctica, se llevó a cabo el control de la temperatura de una planta utilizando un sensor PT100. El sistema se controla mediante la regulación del calefactor (simulado con focos) y un ventilador con velocidades variables, ajustadas según los rangos de temperatura medidos. En la Fig. 10, se muestra el ensamble de la planta acondicionada para su funcionamiento, con el sensor PT100 ubicado en su interior.

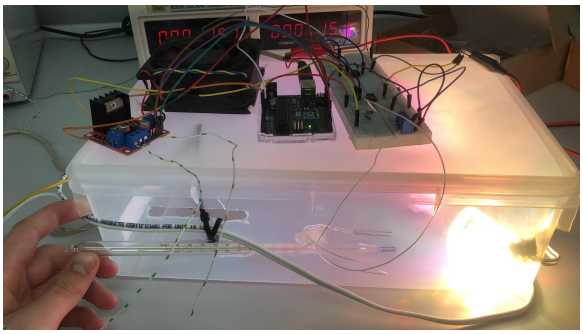


Fig. 10. Planta de control de temperatura

La primera tarea requerida para el desarrollo de esta instancia fue el adecuado acondicionamiento de la PT100, debido a que sus pequeños cambios de resistencia dificultan la medición directa. Tras el análisis de distintas opciones, se decidió implementar un puente de Wheatstone, seguido de una configuración de restador y un amplificador no inversor. El puente de Wheatstone convierte los cambios de resistencia de la PT100 en una señal de voltaje estable. A continuación, el restador amplifica esta pequeña diferencia de voltaje, mientras que el amplificador no inversor incrementa aún más la señal sin invertirla, asegurando que sea suficiente para ser procesada correctamente. Esta configuración se ajustó utilizando resistencias adecuadas para resolver la problemática planteada, logrando que la señal generada por la PT100 sea amplificada y acondicionada para ser medida por el ADC del Atmega328p.

La implementación del acondicionamiento del sensor de temperatura se muestra en la Fig. 11.

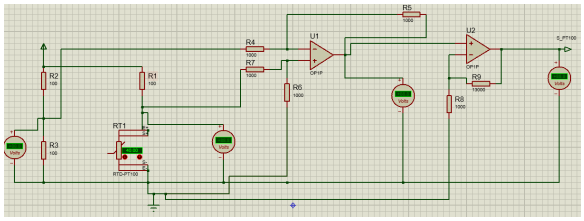


Fig. 11. Circuito de acondicionamiento para la PT100

Tras el acondicionamiento adecuado del sensor, el siguiente paso fue su caracterización, con el fin de determinar una función que permitiera obtener la temperatura registrada por la PT100 según el voltaje medido por el Atmega328p a través de su ADC. Para este proceso, se utilizó un termómetro de alcohol para registrar diferentes temperaturas y sus voltajes correspondientes, con el objetivo de ajustar la función que mejor se adaptara a los valores obtenidos, conocida teóricamente como una función lineal. Para realizar las mediciones, se utilizó una botella de agua, variando la temperatura del agua y colocando tanto la PT100 como el termómetro en su interior, como se observa en la Fig. 12.

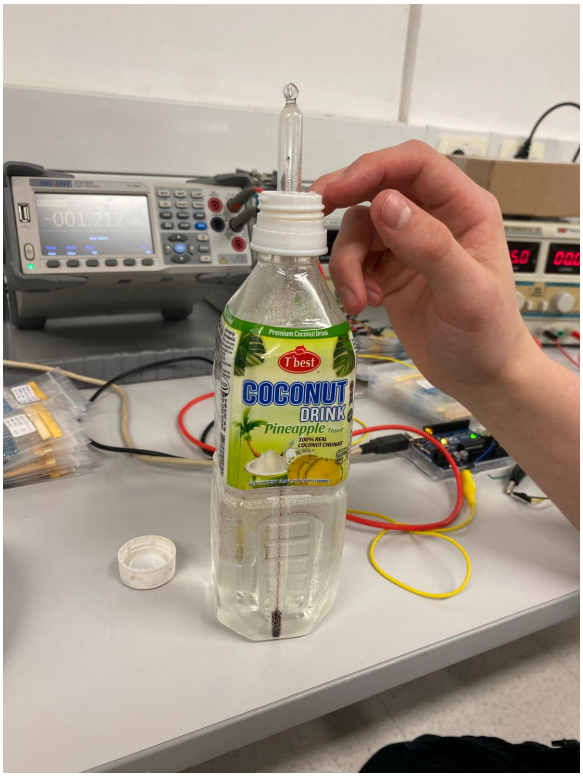


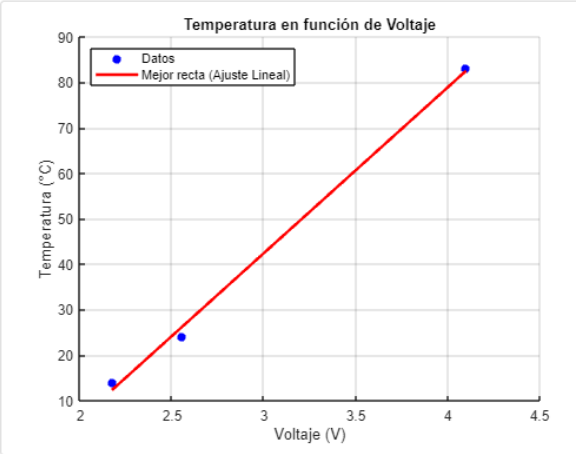
Fig. 12. Proceso para la caracterización de la PT100

Los valores obtenidos se muestran en la Tabla 2.

Temperatura (°C)	Voltaje (V)
14	2.175
24	2.555
83	4.100

TABLE II
VALORES DE TEMPERATURA Y VOLTAJE DETERMINADOS

Tras la recopilación de los datos, se utilizó el software Matlab para graficar la temperatura en función del voltaje, obteniendo la mejor recta de ajuste a los datos. Esto permitió obtener la función que describe la temperatura medida por el sensor en función del voltaje registrado por el ADC. El gráfico generado se muestra en la Fig. 13.




```
Coeficientes de la recta:
Pendiente: 36.5205
Intercepto: -67.1586
```

Fig. 13. Gráfica de temperatura vs voltaje (PT100)

La función de temperatura en función del voltaje, obtenida tras el ajuste, es:

$$T(V) = 36.520V - 67.159$$

Posteriormente, se registraron más valores de temperatura para verificar que la función obtenida se ajusta a cualquier valor de voltaje medido. Los valores registrados se muestran en la Tabla 3.

Temperatura Real(°C)	Voltaje (V)	Temperatura Calculada(°C)
36	2.820	35.83
47	3.140	47.51

TABLE III
VALORES DE TEMPERATURA REALES Y CALCULADOS

Como se puede observar en la Tabla 3, la función obtenida se ajusta correctamente a los datos registrados, presentando un error despreciable para el control de la planta. Además, utilizando Matlab, se generó un gráfico que representa todos los puntos obtenidos, confirmando el comportamiento lineal del sensor PT100 y el correcto proceso de caracterización. Este gráfico se presenta en la Fig. 14.

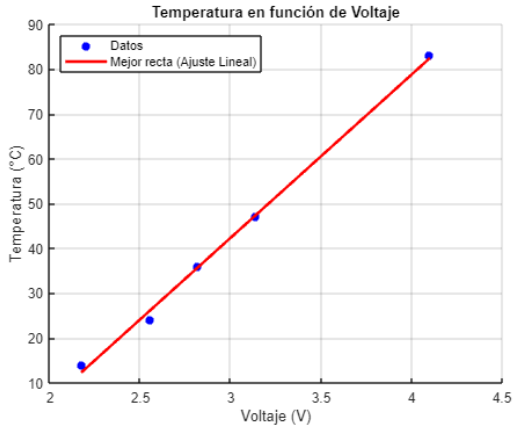


Fig. 14. Gráfica de comprobación y validación de temperatura vs voltaje (PT100)

Los códigos de Matlab utilizados para generar los gráficos pueden consultarse en la plataforma GitHub en el siguiente enlace:

Códigos de Matlab para gráficos.

Una vez obtenida la función que permite determinar la temperatura en función de los voltajes medidos, se avanzó en el desarrollo de la actividad práctica. En este punto, fue posible medir correctamente los valores de temperatura del sistema, permitiendo generar el código de control que ajusta el sistema a los distintos rangos de temperatura requeridos. Para el correcto funcionamiento de la planta, también se realizó la configuración del módulo L298N para controlar la velocidad del ventilador, y la conexión del relé que controla el calefactor. Durante la validación de esta actividad, se compararon constantemente los valores obtenidos por la PT100 con los registrados por el termómetro. Además, se encendió el calefactor para aumentar la temperatura a más de 50°C, habilitando el sistema de control para exponer la planta a todos los rangos de temperatura establecidos en la actividad. El sistema también incluyó la correcta comunicación mediante UART, permitiendo el monitoreo en tiempo real de la planta y la visualización en pantalla de los gráficos solicitados. El usuario tiene la opción de ajustar el punto de temperatura objetivo del sistema, logrando un control total sobre la gestión térmica de la planta.

Para mayor comprensión, se adjuntan enlaces a GitHub con el código del Problema B y su video explicativo.

Código Problema B

Video Problema B

Finalmente, se debe destacar que el proceso de acondicionamiento y caracterización de la PT100 presentó varias dificultades, principalmente debido a la inestabilidad del sensor y los desafíos para su correcto acondicionamiento. Por ejemplo, al medir la temperatura ambiente (24°C según el termómetro) con un voltímetro digital, la PT100 registró un rango de 35 a 40°C, lo que sugiere un funcionamiento errático. Por esta razón, es aceptable que existan algunos márgenes de error en las mediciones. A pesar de las complicaciones, el sistema desarrollado ofrece un control de temperatura óptimo para la planta.

Problema C

En esta sección se presentan los resultados obtenidos tras la implementación del código correspondiente al Problema C.

El sistema debía monitorear y ajustar la velocidad de un motor controlado por PWM, utilizando dos potenciómetros: uno de referencia y otro acoplado al motor. El objetivo principal era igualar el valor de ambos potenciómetros, utilizando el valor del PWM y ajustando el sentido de giro del motor.

El código desarrollado incluye las funciones necesarias para cumplir con los requerimientos del sistema. En primer lugar, se implementó la configuración del ADC para leer los valores de los dos potenciómetros. El potenciómetro de referencia define el valor objetivo, mientras que el potenciómetro del motor mide la posición actual del mismo. Los valores leídos se utilizan para calcular el error entre ambos, que es el valor de PWM aplicado al motor. Además, se desarrollaron funciones para el control de los motores en sentido horario y antihorario, basándose en el error calculado.

La comunicación serial también fue implementada, lo que permitió al sistema informar en tiempo real los valores de referencia, los valores del potenciómetro del motor, el PWM aplicado y el sentido de giro. Esta información fue clave para monitorear el comportamiento del sistema y asegurarse de que los ajustes fueran precisos.

Una de las limitaciones que se encontró durante las pruebas fue que, si variamos muy poco el potenciómetro de referencia, el motor no reaccionaba. Ante esto, se decidió sumar un valor constante al PWM, para asegurar que el potenciómetro del motor lograra corregir correctamente la diferencia entre ambos.

Sin embargo, si el valor sumado era demasiado grande, el motor se movía más de lo necesario, generando un comportamiento no deseado. Por otro lado, si el valor era demasiado pequeño, el problema persistía: el motor no se movía cuando el cambio en el potenciómetro de referencia era mínimo. Luego de realizar varias pruebas iterativas, se encontró un valor óptimo que permitió un funcionamiento correcto del sistema.

A pesar de las limitaciones mencionadas, el sistema pudo cumplir con todos los objetivos propuestos. El motor ajustaba su velocidad y sentido de giro según las diferencias entre los valores de los potenciómetros, y los resultados eran monitoreados en tiempo real mediante la comunicación serial. Además, el comportamiento del sistema fue graficado utilizando Python, Cabe destacar que el gráfico no se genera en tiempo real, sino que se actualiza cada 100 valores registrados, permitiendo visualizar el rendimiento del motor y los valores de referencia y PWM a lo largo del tiempo.

Ahora se presentarán dos pruebas realizadas para evidenciar el correcto funcionamiento del sistema en la igualación de los valores de los potenciómetros.

En la primera prueba, se dejó el potenciómetro de referencia en un valor determinado mientras el sistema permanecía sin alimentación. Posteriormente, al activar el sistema, se observó

si el valor del potenciómetro acoplado al motor se ajustaba al del potenciómetro de referencia. Tal como se muestra en el gráfico de la Fig. 15, inicialmente el valor del potenciómetro de referencia se mantiene constante. Una vez que el sistema recibe alimentación, representado por la variación en la línea roja, el valor del potenciómetro del motor comienza a ajustarse hasta casi igualarse con el valor de referencia, cumpliendo con el comportamiento esperado.

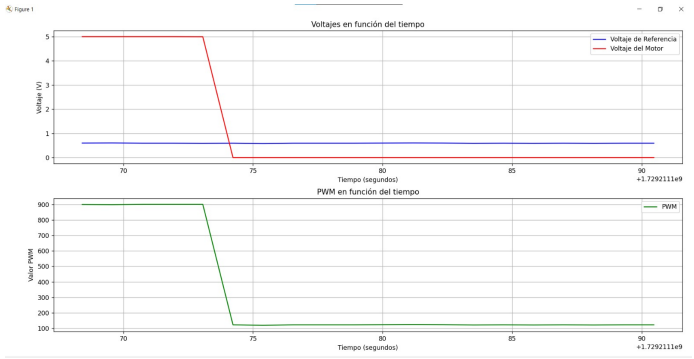


Fig. 15. Gráfica Python Problema D - Prueba 1

En la segunda prueba, con el sistema ya alimentado, se varió el valor del potenciómetro de referencia, llevándolo desde su valor máximo de voltaje hasta su valor mínimo. En esta ocasión, se observó cómo el potenciómetro del motor seguía el comportamiento del de referencia con gran precisión, presentando una mínima diferencia en los tiempos de ajuste. Este comportamiento también puede apreciarse claramente en el gráfico de la Fig. .

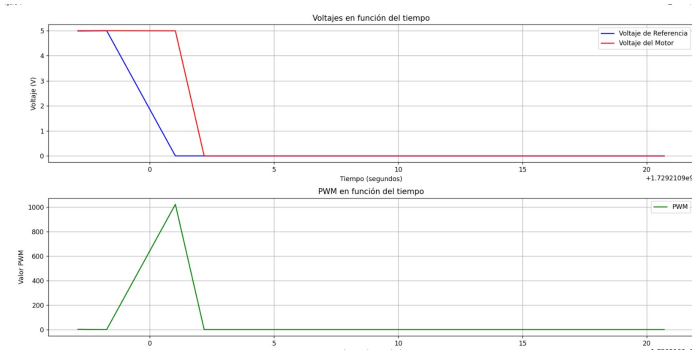


Fig. 16. Gráfica Python Problema D - Prueba 2

Adicionalmente, los resultados se pueden visualizar tanto en el código, como en un video subido a la plataforma GitHub, disponibles en los siguientes enlaces:

Enlace: ***Código Problema C***.

Enlace: ***Video Problema C***.

Problema D

En esta sección se presentan los resultados obtenidos tras la implementación del código correspondiente al Problema D. Para la última problemática del informe, como primera medida para abordar su solución, se realizó una simulación particular con el fin de evaluar el código que se implementaría en la práctica.

Al emplear una fotorresistencia (LDR), resultó necesario el uso del puente de Wheatstone para obtener el voltaje en función de la resistencia variable del sensor. Dependiendo del color del papel, el puente de Wheatstone entregaba un valor de voltaje específico. Este valor fue medido utilizando un multímetro digital, con el propósito de definir correctamente los rangos en el código, de manera que este actuara de acuerdo con dichas condiciones. Gracias a la lectura del convertidor analógico-digital (ADC) del sensor LDR, se pudo obtener un valor de voltaje en un rango de 0 a 5 V, multiplicando por 5 y dividiendo por 1023 el valor ADC obtenido.

Es importante aclarar que los colores seleccionados para la lectura fueron celeste, violeta y rosado. La razón por la que se eligió esta paleta de colores radica en la diferencia notable entre los voltajes resultantes. En comparación con otra selección de colores, esta elección facilitó la definición de los rangos de voltaje.

Para asegurar una iluminación constante, se implementó un LED blanco, lo que ayudó a evitar problemas relacionados con la variación de la luz ambiental, la cual podría afectar significativamente el sistema implementado. Una vez definidos todos los rangos, se procedió al desarrollo del sistema, es decir, la comunicación con el usuario a través de UART y el control del movimiento del servomotor.

Con el valor de voltaje leído desde el pin analógico 0 del LDR, se implementó la comunicación UART para cumplir con este requerimiento de la práctica. Adicionalmente, al solicitarse la diferencia entre el valor establecido y el valor de lectura, se realizó una resta entre el valor leído y los valores mínimo y máximo correspondientes al color que se estaba leyendo, y el resultado fue enviado a través de UART.

Finalmente, mediante la configuración del modo fast PWM, se implementó el control del servo, que se posiciona en un ángulo determinado al recibir diferentes valores enviados al registro OCR1A, lo que indica el color leído. Cabe destacar que se realizaron múltiples pruebas con diversos valores para lograr orientar correctamente el servo, encontrando los valores más adecuados, los cuales se detallan en la tabla 4.

Ángulo del Servo (°)	**Valor OCR1A**	**Colores**
0°	5300	Celeste
90°	3300	Violeta
180°	1500	Rosado

TABLE IV
ÁNGULOS DEL SERVO, VALORES DE OCR1A Y COLORES ASOCIADOS

Adicionalmente, los resultados se pueden visualizar tanto en el código, como en un video subido a la plataforma GitHub, disponible en el siguiente enlace.

Enlace: ***Código Problema D***.

Enlace: ***Video Problema D***.

VIII. Conclusiones

Al finalizar este laboratorio, se puede concluir que los objetivos establecidos al inicio fueron alcanzados con éxito. El propósito principal era implementar sistemas de control utilizando el microcontrolador ATmega328P para resolver problemas relacionados con el movimiento de un plotter, el control de motores, la regulación de temperatura y la detección de colores, y se cumplió de manera efectiva.

En la Parte 1, se profundizó en el funcionamiento del plotter, logrando su control preciso mediante relés y asegurando la ejecución adecuada de figuras geométricas predefinidas. A través del análisis y pruebas iterativas, se comprendió el ajuste de tiempos de activación de los motores, lo que permitió superar las limitaciones iniciales y lograr una mayor flexibilidad en los movimientos.

En la Parte 2, se resolvieron con éxito los cuatro problemas planteados. En el Problema A, se implementó un sistema de control interactivo para el plotter, que facilitó la selección y ejecución de figuras mediante comunicación serial. En el Problema B, se desarrolló un sistema de control automático de temperatura utilizando un sensor PT100, asegurando la regulación precisa del ambiente mediante el uso de un calefactor y un ventilador de velocidad variable. En el Problema C, se implementó un sistema de control de motor basado en la lectura de dos potenciómetros, ajustando su velocidad y dirección mediante señales PWM. Finalmente, en el Problema D, se diseñó un sistema de detección de colores que controlaba un servomotor, permitiendo su posicionamiento en función del color detectado.

En resumen, el laboratorio permitió afianzar el conocimiento sobre el uso del microcontrolador ATmega328P en aplicaciones prácticas, logrando implementar soluciones efectivas para cada problema planteado y demostrando el éxito de cada una de las etapas del laboratorio.

IX. Referencias bibliográficas

[1] S. F. Barrett, *Embedded system design with the Atmel AVR microcontroller: Part I*, vol. 24. Morgan & Claypool Publishers, 2009. doi: 10.2200/S00225ED1V01Y200910DCS025.

[2] S. F. Barrett, *Embedded system design with the Atmel AVR microcontroller: Part II*, vol. 25. Morgan & Claypool Publishers, 2009. doi: 10.2200/S00225ED1V01Y200910DCS025.

[3] S. F. Barrett and D. J. Pack, *Atmel AVR microcontroller primer: Programming and interfacing*, 2nd ed. Morgan & Claypool, 2012.

[4] S. P. Dandamudi, *Guide to RISC processors for programmers and engineers*, 1st ed. Springer New York, 2005. doi: 10.1007/b139084.

[5] D. M. Harris and S. L. Harris, *Digital design and computer architecture*. Morgan Kaufmann, 2007.

[6] W. Kühnel, *AVR RISC microcontroller handbook*, 1st ed. Newnes, 1998.

[7] A. N. Sloss, D. Symes, and C. Wright, *ARM system developer's guide: Designing and optimizing system software*, 1st ed. Elsevier/Morgan Kaufmann, 2004.