

# Tecnología en Microprocesamientos: Taller 2

Andrés Nicolás Gonzalez Lopez

*Ingeniería en Mecatrónica*

*Universidad Tecnológica (UTEC)*

Fray Bentos, Uruguay

andres.gonzalez.l@estudiantes.utec.edu.uy

Jerónimo Fernández González

*Ingeniería en Mecatrónica*

*Universidad Tecnológica (UTEC)*

Fray Bentos, Uruguay

jeronimo.fernandez@estudiantes.utec.edu.uy

Priscila Rossi Ferrari

*Ingeniería en Mecatrónica*

*Universidad Tecnológica (UTEC)*

Fray Bentos, Uruguay

priscila.rossi@estudiantes.utec.edu.uy

## RESUMEN

Este laboratorio se dividido en cuatro partes principales. En la primera parte, se implementó el control de un plotter para dibujar figuras geométricas y otras formas como un ratón y un conejo a través de comunicación serial. La segunda parte abordó el control automático de temperatura utilizando un sensor PT100, donde el sistema gestiona un calefactor y un ventilador según el valor medido, con opciones de ajuste dinámico a través de un menú interactivo. En la tercera parte, se controló un motor utilizando potenciómetros para ajustar su velocidad y dirección mediante modulación por ancho de pulso (PWM), y se monitoreó el sistema en tiempo real con un código en Python. Finalmente, la cuarta parte consistió en un sistema de selección de colores utilizando una fotocelda y un servomotor, el cual detectaba colores en una hoja de referencia y posicionaba el servomotor según el color identificado.

## INTRODUCCIÓN

Este laboratorio se divide en cuatro partes principales, en las cuales se aborda diferentes aspectos del control y monitoreo de sistemas macatrónicos.

En la primera parte, se estudió el funcionamiento básico de un plotter , para implementar un código que, por medio del puerto serial del ATmega328P, controle al plotter para dibujar figuras geométricas predefinidas, como un triángulo, un círculo y un cuadrado. Además, el equipo trabajó con dos figuras la cuales son un ratón y un conejo.

En el segundo problema, se desarrolló un sistema de control automático de temperatura utilizando un sensor PT100. El sistema fue programado para tomar decisiones en función de la temperatura medida, controlando un calefactor y un ventilador a distintas velocidades para mantener la temperatura ambiente dentro de un rango. También, se implementó un menú interactivo que permitía ajustar el "punto medio" de la temperatura, recalibrando automáticamente los umbrales superiores e inferiores.

El tercer problema del laboratorio se centró en el control de un motor utilizando dos potenciómetros: uno de referen-

cia y otro acoplado al eje del motor. El objetivo fue programar el ATmega328P para ajustar la velocidad y dirección del motor, de modo que el valor del potenciómetro en el eje del motor se igualara al valor del potenciómetro de referencia. Se implementó un sistema de control por modulación por ancho de pulso (PWM) para ajustar la velocidad del motor. Los parámetros del sistema, incluyendo los valores de los potenciómetros y el PWM, se monitorearon en tiempo real mediante la implementación de un código en Python.

En el problema D, se desarrolló un sistema de selección de colores utilizando una fotocelda y un servomotor, controlados por el ATmega328P. El sistema fue diseñado para detectar colores presentes en una hoja de referencia y mover el servomotor a una posición determinada en función del color identificado. Los valores de la fotocelda, el color detectado y el ángulo correspondiente del servomotor se visualizan mediante comunicación serial.

## OBJETIVO GENERAL

Controlar, mediante el Atmega328P y programación en C, el funcionamiento de un Plotter, un sistema de monitoreo de la temperatura, un sistema de manejo de un motor mediante potenciómetros y un sistema de selección de colores con una foto-resistencia y un servomotor.

## OBJETIVOS ESPECÍFICOS

### Problema A

- Dibujar un cuadrado, un triángulo, un circulo , un ratón y un conejo en el plotter.
- Mostrar un menú interactivo a través de puerto serial.

### Problema B

- Medir y controlar la temperatura ambiente de manera automática usando el ATmega328P y el sensor PT100.
- Simular y verificar el correcto funcionamiento del control de temperatura en Proteus.
- Implementar físicamente el sistema para controlar el calefactor y el ventilador, manteniendo la temperatura dentro de los rangos definidos.

- Permitir ajustes dinámicos del punto medio de temperatura mediante un menú interactivo.
- Monitorear en tiempo real la evolución de la temperatura y las acciones del sistema tanto en simulación como en la práctica física.

#### Problema C

- Leer el valor de un potenciómetro que actúa como referencia.
- Ajustar el giro del motor para igual el valor de un segundo potenciómetro, al de referencia.
- Utilizar modulación por ancho de pulso para variar la velocidad del motor.

#### Problema D

- Detectar el color presente en una hoja de referencia utilizando una foto-celda.
- Posicionar un servomotor en un ángulo determinado en función del color identificado.

#### METODOLOGÍA

- Problema A: Para abordar la problemática de esta primera parte fue necesario entender el funcionamiento del plotter. En base a los avisos y la información que se brindó en el documento de laboratorio se desarrolló el código para el mismo. Con la imagen 2, que se encuentra en anexos, se probó el movimiento del plotter y se comenzó el desarrollo del código, el cual debió ajustarse al tiempo de conmutación de los relés. Una vez que se logró ajustar el movimiento del sistema de la manera que se quería, se realizaron las primeras figuras, las cuales fueron el cuadrado, el triángulo y el círculo respectivamente. A partir de estas figuras tres figuras se dibujó la figura del conejo y del ratón por separado, para finalmente implementar la comunicación serial y un menú que es capaz de controlar el movimiento de plotter, así como ordenarle realizar alguna de las figuras ya mencionadas.

- Problema B:  
En esta parte de la práctica, se desarrolló un sistema de control de temperatura utilizando el microcontrolador ATmega328P y un sensor de temperatura PT100. El diseño inicial y la simulación del sistema se realizaron en el software Proteus, donde el comportamiento fue simulado de manera exitosa. El sistema fue programado en lenguaje C para realizar mediciones de temperatura cada 5 segundos, y controlar un calefactor y un ventilador según los siguientes rangos de temperatura:

- 0-22°C: Encender el calefactor para aumentar la temperatura.
- 23-30°C: Mantener el calefactor y el ventilador apagados (rango óptimo).
- 31-40°C: Activar el ventilador a baja velocidad.
- 41-50°C: Activar el ventilador a velocidad media.
- Más de 51°C: Activar el ventilador a alta velocidad para prevenir sobrecalentamientos.

El sistema también fue diseñado para permitir el ajuste dinámico del punto medio de la temperatura mediante un menú interactivo que recalibra los rangos superiores e inferiores automáticamente. La simulación fue validada en Proteus, sin embargo, durante la implementación física, surgieron problemas relacionados con la amplificación de la señal del sensor PT100. Se utiliza un amplificador de instrumentación para acondicionar la señal medida de modo que se obtenga una salida de 0 a 5 V al medir de 0 a 100 °C.

#### I. CARACTERIZACION PT100

Para este caso, los valores obtenidos en voltaje del puente de Wheatstone fueron iguales a los de la simulación. Sin embargo, hubo inconvenientes en la amplificación, la cual se realizó con amplificadores LM741 en configuración de instrumentación, lo que resultó en que la variación en el voltaje de salida fuera casi insignificante. Esto generó problemas al intentar caracterizar correctamente la PT100. Por lo tanto, se elaboró una gráfica con valores ideales para demostrar el conocimiento necesario para realizar la linealización y obtener la ecuación de la recta, como se muestra en la figura 4

- Problema C: Para este tercer problema se inició con el desarrollo de un código en C que fuese capaz de medir la diferencia de potencia entre dos potenciómetros en un rango de 0-1023. Para comprobar el funcionamiento del mismo se hizo una simulación en Proteus para avanzar en el proceso del código. Como se simuló se puede ver en la figura 6 (ubicada en los Anexos). Se utilizó un puente H L298n para controlar un motor DC. Una vez que se verificó que el código envía, mediante puerto serial, los valores de PWM y de los potenciómetros de manera correcta en la simulación, se pasó a realizar los ajustes necesarios en la implementación en físico. En el entorno de programación de Visual Studio Code, mediante un código en Python y utilizando la librería Pyserial, se graficó en tiempo real los valores de ambos potenciómetros, del PWM y de la diferencia entre estos dos.

- Problema D:

Se utilizó un servomotor ubicado sobre un círculo con 3 colores determinado, este estaba conectado al pin PB1 del microcontrolador ATmega328P para posicionar su eje según el color detectado por una fotocelda. Se implementó la comunicación UART para monitorear los valores del sensor de luz, el color identificado y los ángulos de posición del servomotor. Para ello, se configuraron los registros del microcontrolador para habilitar el ADC en el canal ADC0, con referencia de voltaje AVcc y un prescaler de 64. La fotocelda se conectó al pin ADC0 (PC0), permitiendo realizar la lectura analógica del valor de luz, y el servomotor se controló con señales PWM generadas por el Timer 1, con el pin PB1 configurado como salida. La parte

principal consistió en calibrar la fotocelda para identificar tres colores: verde, amarillo y rojo, con valores de ADC que abarcan los rangos de 0-340 para verde, 341-681 para amarillo y 682-1023 para rojo. A través de la función `USART_sendString`, se envió al monitor serial el valor ADC leído y el color detectado. Además, se implementó la función `servoMove` para ajustar la posición del servomotor en tres ángulos específicos: 180 grados para verde, 90 grados para amarillo y 0 grados para rojo, utilizando valores de OCR1A de 850, 2500 y 4500 respectivamente, para controlar el ancho de pulso PWM. Se configuró el UART a 9600 baudios para transmitir los datos al terminal serial, como el valor de la fotocelda, el color detectado y el ángulo del servomotor.

## II. RESULTADOS

### A. Funcionamiento y Control del Plotter

Para el plotter se logró completar las figuras solicitadas y se pueden visualizar en la figura 1, pero para ver el proceso que le toma al plotter hacer estos dibujos, se puede encontrar en el link de los anexos un video tomado del momento en el que el plotter dibuja estas figuras.

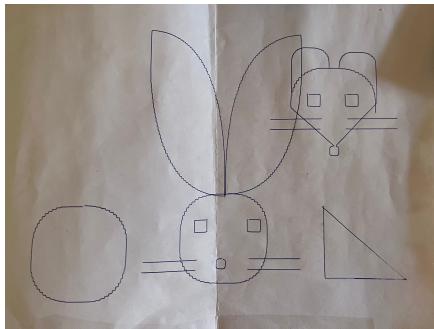


Fig. 1. Figuras hechas por el Plotter

### B. Sistema de monitoreo de la temperatura

A pesar de que la simulación fue perfecta en Proteus, se presentaron complicaciones en la implementación física, específicamente en la amplificación de la señal del sensor PT100. Se utilizó una configuración de amplificación de instrumentación, pero esta no fue capaz de amplificar la señal adecuadamente, lo que impidió que el microcontrolador pudiera procesar correctamente las mediciones de temperatura.

Este fallo en la amplificación de la señal afectó directamente la operación de los dispositivos controlados (calefactor y ventilador), impidiendo que el sistema funcionara correctamente en un entorno físico. Como resultado, no fue posible completar la práctica en su totalidad.

#### 1) Cálculos Medición de Temperatura con PT100:

**Datos Iniciales:** Para una temperatura de 0 °C, se tiene que:

$$R_x = 100 \Omega \Rightarrow V_B - V_A = 0$$

Por lo tanto,  $R_x = R_3 = 100 \Omega$ .

Para una temperatura de 100 °C:

$$R_x = 138.5 \Omega$$

De acuerdo a la tabla de datos de PT100, se realizan los cálculos:

*Cálculo de Voltajes:*

$$V_A = \frac{5 \times 100}{10k + 100} = 49.5 \text{ mV}$$

$$V_B = \frac{5 \times 138.5}{10k + 138.5} = 68.3 \text{ mV}$$

$$\Delta V = V_B - V_A = 18.8 \text{ mV}$$

*Ecuaciones del Amplificador:*

$$V_0 = (V_\beta - V_\alpha) \frac{R_3}{R_2}$$

$$V_\beta - V_\alpha = V_0 \frac{R_2}{R_3}$$

$$V_\alpha - V_\beta = -V_0 \frac{R_2}{R_3}$$

*Relación entre Voltajes:*

$$\frac{V_A - V_B}{R_G} = \frac{V_\alpha - V_\beta}{R_G} + 2R_1$$

Sustituyendo y realizando álgebra:

$$V_A - V_B/R_G = -(V_0 \frac{R_2}{R_3}) \left( \frac{1}{R_G} + 2R_1 \right)$$

Para concretar la condición 5 V en  $V_0$ , se eligen:

$$R_2 = R_3 = 40 \text{ k}\Omega \quad \text{y} \quad R_1 = 25 \text{ k}\Omega$$

Sustituyendo en la ecuación:

$$\frac{V_A - V_B}{R_G} = - \left( V_0 \frac{R_2}{R_3} \right) \left( \frac{1}{R_G} + 2R_1 \right)$$

Se deduce que:

$$R_G = \frac{2R_1}{\left( \frac{V_0}{\Delta V} \frac{R_2}{R_3} \right)} - 1$$

Por lo tanto, calculando:

$$R_G = 188.7 \Omega$$

### C. Manejo de un motor mediante potenciómetros

El video del funcionamiento, como los códigos utilizados para esta implementación se encuentran en el link a git en los anexos. En los mismos también se puede ver en la figura 7 como se conectó el sistema cuando se llevó a cabo su implementación en físico

#### D. Sistema de selección de colores

Se cumplieron parcialmente las peticiones para esta etapa por el hecho que en la simulación se visualiza con un rango determinado cual es el color 1, correspondiente al rojo, color 2 siendo el amarillo y por ultimo el color 3 siendo el verde. Pero en el momento de llevarlo a la práctica solo detecta correctamente el color verde (3) ya que al acercar el círculo cromático con el tono del mismo la aguja del servomotor pasa al ángulo 180 pero no es posible visualizarlo en la terminal serial.

### III. CONCLUSIÓN

El laboratorio permitió explorar varias aplicaciones de la capacidad del ATmega328P para manejar tareas como el control de un plotter, la regulación de temperatura, el manejo de motores mediante potenciómetros y la detección de colores. Aunque algunos sistemas funcionaron correctamente en simulación, surgieron dificultades en la implementación física, como problemas en la amplificación de la señal del sensor PT100 y en la detección precisa de colores por la fotocelda. A pesar de esto, se obtuvieron buenos resultados a nivel de programación y control de sistemas y se validaron varias de las funcionalidades propuestas.

### BIBLIOGRAFÍA

- <https://www.alldatasheet.com/html-pdf/170960/FAIRCHILD/LM741/484/1/LM741.html>
- <https://www.microchip.com/wwwproducts/en/ATmega328P>

### ANEXOS

Repositorio de Lab2 en github

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

Fig. 2. Pines para control de Plotter

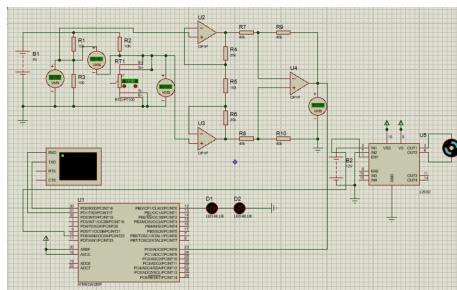


Fig. 3. Simulación Circuito Problema B

voltaje (v)	temperatura (C)
1.87	24
2.1	30
2.28	35
2.45	40
2.61	45
2.8	50
3	55
3.15	60
3.32	65
3.47	70
3.65	75
3.78	80

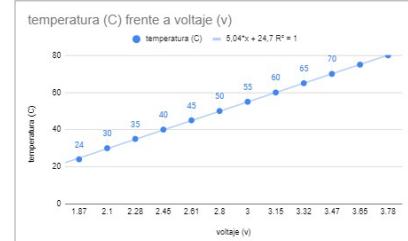


Fig. 4. Caracterización Pt100

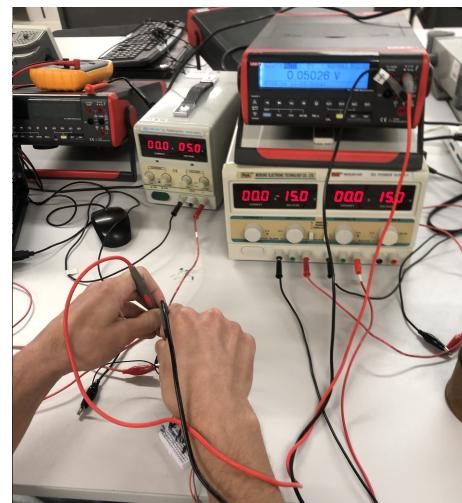


Fig. 5. Puente de Wheatstone

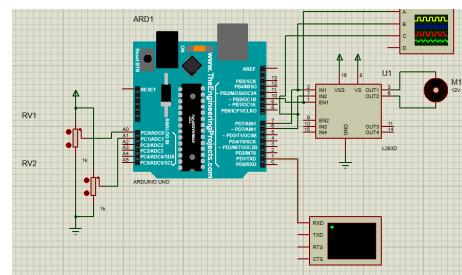


Fig. 6. Simulación Circuito Problema C

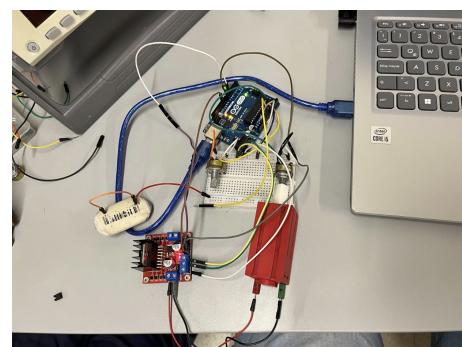


Fig. 7. Circuito Problema C

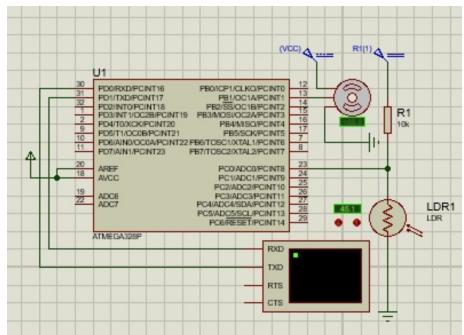


Fig. 8. Simulación Circuito Problema D

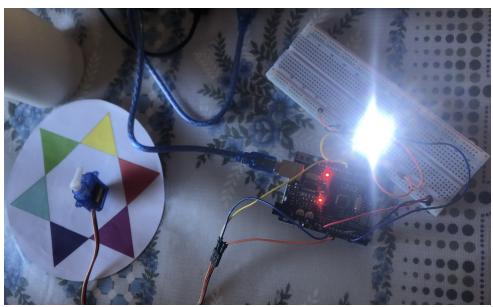


Fig. 9. Circuito Problema D