

# Tercera práctica de Laboratorio

\*Tecnologías de Microprocesamiento

1<sup>nd</sup> Marcos Casanova

Universidad Tecnológica del Uruguay

Fray Bentos, Río Negro, Uruguay

marcos.casanova@estudiantes.utec.edu.uy

**Abstract—**  
**Index Terms—**

## I. INTRODUCCIÓN

*Objetivo General:*

Diseñar, programar e implementar diversos sistemas embebidos basados en el microcontrolador ATmega328P, aplicando los principios de adquisición de datos, control de actuadores y comunicación serial para resolver problemáticas prácticas de la ingeniería mecatrónica, integrando tanto simulaciones como prototipos físicos que demuestren el funcionamiento, la estabilidad y la interacción entre hardware y software en entornos de control y automatización.

*Objetivos Específicos*

- Implementar el control del plotter electrónico mediante el microcontrolador ATmega328P, gestionando motores paso a paso y una válvula solenoide a través de comunicación serial, para la ejecución de figuras geométricas predefinidas.
- Diseñar y programar un sistema de control de temperatura utilizando un sensor LM35, aplicando control automático con calefactor y ventilador, y permitiendo el ajuste dinámico del punto medio de operación por UART.
- Desarrollar un sistema de control de posición de motor basado en la lectura de dos potenciómetros y modulación PWM, con retroalimentación analógica que permita igualar una referencia establecida.
- Construir un sistema de matriz LED RGB controlado mediante joystick, integrando lecturas analógicas del ADC para el desplazamiento del LED activo y cambios de color aleatorios mediante interrupción o pulsador. una cerradura electrónica RFID que utilice el módulo RC522 y memoria EEPROM para registrar, verificar, actualizar y eliminar credenciales de acceso, con visualización de estado en pantalla LCD y comunicación UART.

## II. MARCO TEÓRICO

*A. Memoria EEPROM (Electrically Erasable Programmable Read-Only Memory)*

La memoria EEPROM es un tipo de almacenamiento no volátil que permite escribir, borrar y conservar información incluso en ausencia de energía eléctrica. A diferencia de la ROM convencional, sus celdas pueden reprogramarse

eléctricamente, y a diferencia de la memoria Flash, las operaciones de escritura se realizan a nivel de byte individual, ofreciendo flexibilidad en el manejo de configuraciones o parámetros del sistema.

En los microcontroladores AVR, como el ATmega328P, la EEPROM se gestiona mediante los registros EEAR, EEDR y EECR, con tiempos de escritura del orden de los milisegundos y una durabilidad aproximada de  $10^5$  ciclos por celda. Este tipo de memoria se utiliza en contextos donde se requiere persistencia de datos configurables, como identificadores de usuario, calibraciones o contraseñas.

[Microchip Technology Inc., 2020; Atmel Corporation, 2009]

*B. Conversión Analógica-Digital (ADC)*

Un conversor analógico-digital (ADC) es un circuito que transforma una señal analógica continua en una representación numérica discreta, permitiendo su procesamiento digital. Los ADCs se caracterizan por su resolución (número de bits) y su velocidad de muestreo.

El ATmega328P incorpora un ADC de 10 bits con hasta 8 canales multiplexados, capaz de medir tensiones entre 0 y 5 V con una resolución de aproximadamente 4,88 mV por bit. Internamente, el proceso se basa en un convertidor de aproximaciones sucesivas (SAR) que compara la tensión de entrada con valores de referencia.

El uso adecuado del prescaler del reloj y la elección de la referencia de tensión (AREF o Vcc) son determinantes para lograr precisión y estabilidad. Este tipo de conversores se encuentra en aplicaciones de sensado como temperatura, posición o intensidad lumínica.

[Horowitz & Hill, 2015; Microchip, 2021]

*C. Interfaz I<sup>2</sup>C (Inter-Integrated Circuit)*

El bus I<sup>2</sup>C, desarrollado por Philips en los años 80, es un estándar de comunicación serie síncrona de dos hilos (SDA para datos y SCL para reloj) que permite conectar múltiples dispositivos maestros y esclavos en una sola línea.

Cada componente se identifica por una dirección única, y la comunicación se basa en condiciones de inicio/paro y confirmación mediante bits de reconocimiento (ACK/NACK). Su simplicidad y bajo consumo de pines lo han convertido en uno de los protocolos más empleados en sistemas embebidos para integrar sensores, pantallas o memorias externas.

En los microcontroladores AVR, el módulo TWI (Two-Wire Interface) implementa el protocolo I<sup>2</sup>C a nivel de hardware, soportando velocidades de hasta 400 kHz en modo Fast Mode.

[NXP Semiconductors, 2014; IEEE Std 1194-1992]

#### D. Protocolo SPI (Serial Peripheral Interface)

El protocolo SPI, diseñado por Motorola, ofrece una comunicación serie síncrona full-duplex que se caracteriza por su alta velocidad (hasta varios MHz) y baja complejidad de implementación. Emplea cuatro señales principales: MOSI, MISO, SCK y SS (Slave Select).

La transferencia de datos ocurre simultáneamente entre maestro y esclavo mediante un registro de desplazamiento interno. Es especialmente útil en sistemas que requieren alta velocidad de actualización, como memorias Flash externas, módulos de sensores o lectores RFID.

Su principal desventaja frente a I<sup>2</sup>C es la necesidad de una línea SS independiente por cada esclavo, lo que puede incrementar el cableado en sistemas con múltiples dispositivos.

[Motorola, 1985; Microchip, 2021]

#### E. Pantalla LCD (Liquid Crystal Display)

Las pantallas de cristal líquido funcionan mediante la orientación de moléculas de cristal líquido que modifican la polarización de la luz bajo la influencia de un campo eléctrico. Los módulos controlados por el chip HD44780 son el estándar en instrumentación didáctica y sistemas embebidos, permitiendo mostrar caracteres ASCII en líneas de 16x2 o 20x4.

La interfaz puede implementarse en modo paralelo (4 u 8 bits) o mediante un expansor I<sup>2</sup>C, que traduce los datos al formato requerido por el controlador. El funcionamiento básico implica el envío de comandos (configuración, desplazamiento, borrado) y datos (caracteres) sincronizados por señales RS, RW y E.

El contraste del LCD depende de una tensión variable aplicada al pin V<sub>0</sub>, y su retroiluminación LED requiere control independiente.

[Hitachi, HD44780U Datasheet, 1998; Lim & Kim, 2016]

#### F. Comunicación UART (Universal Asynchronous Receiver-Transmitter)

La UART es un periférico que convierte flujos de datos paralelos en serie y viceversa, sin necesidad de un reloj compartido, mediante el uso de bits de inicio y parada para sincronización.

El formato de transmisión estándar 8N1 (8 bits de datos, sin paridad, 1 bit de parada) es ampliamente compatible con sistemas informáticos y dispositivos embebidos.

Los registros UBRRn, UCSRnA/B/C del ATmega328P permiten ajustar la velocidad de transmisión y los parámetros del frame. Su sencillez y confiabilidad hacen de la UART la interfaz más empleada para depuración, comunicación con PC o intercambio entre microcontroladores.

[Microchip, 2020; Pont, 2002]

#### G. LEDs WS2812

Los diodos WS2812 son LEDs RGB inteligentes que integran un controlador digital en el mismo encapsulado, lo que permite direccionarlos de forma individual a través de un protocolo de un solo hilo ("1-Wire").

Cada LED interpreta secuencias de pulsos con duraciones precisas que codifican los niveles de color en formato GRB. Este protocolo exige una temporización del orden de los 1,25 s por bit, lo que requiere desactivar interrupciones durante la transmisión para evitar distorsiones.

Su arquitectura en cascada permite controlar largas tiras o matrices con un solo pin de datos, siendo ampliamente utilizada en iluminación programable, efectos visuales y visualización dinámica.

[Adafruit Industries, WS2812 Datasheet, 2019; IEEE Spectrum, 2022]

#### H. Identificación por Radiofrecuencia (RFID) – Módulo RC522

La tecnología RFID (Radio Frequency Identification) utiliza campos electromagnéticos para identificar de manera inalámbrica objetos o personas mediante etiquetas (tags) que contienen un chip y una antena.

El módulo RC522, basado en el chip MFRC522 de NXP, opera en la banda de 13.56 MHz y emplea modulación ASK (Amplitude Shift Keying) junto con el estándar ISO/IEC 14443A.

El sistema consta de un lector que genera el campo electromagnético, una etiqueta pasiva que se alimenta de este campo y un proceso de intercambio de datos por acoplamiento inductivo. La comunicación entre el RC522 y el microcontrolador puede realizarse por SPI o I<sup>2</sup>C, y admite

autenticación mediante claves internas.

Este tipo de módulos son frecuentes en sistemas de acceso, control industrial o trazabilidad logística.

[NXP Semiconductors, MFRC522 Datasheet, 2019; ISO/IEC 14443-2:2016]

### *I. Modulación por Ancho de Pulso (PWM)*

La PWM es una técnica de control utilizada para regular energía o velocidad media mediante una señal periódica cuyo ciclo útil (duty cycle) determina la potencia efectiva entregada a una carga.

En el dominio temporal, un ciclo del 100% equivale a alimentación continua, mientras que uno del 0% implica desconexión total. El promedio de tensión o corriente resultante se ajusta proporcionalmente al duty cycle.

Los microcontroladores AVR disponen de temporizadores que pueden operar en modo Fast PWM o Phase Correct, ofreciendo resoluciones de 8 o 16 bits. Esta técnica es fundamental para el control de motores, regulación de temperatura o modulación de brillo en dispositivos LED.

[Erickson & Maksimović, 2001; Microchip, 2021]

### *J. Sensor de Temperatura LM35*

El LM35, desarrollado por Texas Instruments, es un sensor de temperatura analógico de precisión, cuya salida lineal es de 10 mV/°C. No requiere calibración externa y su precisión típica es de  $\pm 0,5$  °C en el rango de 25 °C.

A diferencia de los termistores, el LM35 proporciona una respuesta estrictamente lineal y baja impedancia de salida, facilitando su conexión directa al ADC de un microcontrolador.

Su simplicidad y fiabilidad lo han convertido en un estándar en medición térmica para instrumentación didáctica y control ambiental.

[Texas Instruments, LM35 Datasheet, 2018; Boylestad & Nashelsky, 2019]

### *K. Motores Paso a Paso*

Los motores paso a paso son actuadores electromecánicos que transforman pulsos eléctricos discretos en incrementos angulares fijos, proporcionando control de posición sin retroalimentación.

Su funcionamiento se basa en la excitación secuencial de bobinas que generan campos magnéticos alternantes. Existen configuraciones unipolares y bipolares, siendo estas últimas más eficientes por aprovechar toda la bobina.

La resolución del movimiento depende del número de pasos por revolución y del microstepping aplicado por el controlador. Son esenciales en aplicaciones de precisión como impresoras, robots cartesianos o sistemas de trazado.

[Jones & Flynn, 2013; Hughes & Drury, 2019]

### *L. Potenciómetro*

El potenciómetro es un resistor variable de tres terminales que funciona como divisor de tensión ajustable, entregando una salida proporcional al desplazamiento angular de su eje.

En sistemas analógicos o embebidos, se utiliza tanto como entrada de control (para establecer una referencia) como sensor de posición. Su relación lineal o logarítmica depende del material resistivo y su geometría interna.

Cuando se combina con un ADC, permite digitalizar variaciones mecánicas en señales eléctricas, siendo un dispositivo de gran valor en interfaces hombre-máquina.

[Sedra & Smith, 2016]

### *M. Ventiladores Eléctricos (FAN)*

Un ventilador es un actuador electromecánico rotativo destinado a generar flujo de aire mediante el movimiento de aspas impulsadas por un motor DC.

La velocidad de rotación puede controlarse modulando la tensión de alimentación o mediante técnicas PWM, donde el valor medio del ciclo útil determina la cantidad de energía aplicada.

En ingeniería de control térmico, los ventiladores cumplen un rol crucial en la disipación de calor, favoreciendo la estabilidad de componentes electrónicos y prolongando su vida útil.

[Cengel & Ghajar, 2015; Erickson & Maksimović, 2001]

### *N. Joystick Analógico*

El joystick analógico es un dispositivo de entrada bidimensional que emplea dos potenciómetros ortogonales para detectar desplazamientos en los ejes X e Y, y un interruptor central para eventos discretos (clic).

Los valores analógicos obtenidos se interpretan como coordenadas dentro de un rango definido, lo que permite un control proporcional del movimiento.

El principio de funcionamiento se basa en la variación de resistencia al girar el eje del potenciómetro, generando tensiones entre 0 y 5 V que luego se convierten digitalmente por el ADC del microcontrolador.

Su diseño compacto y su respuesta continua lo hacen ideal para interfaces humanas, robótica y sistemas de control

interactivo.

[IEEE Human Interface Devices Std, 2017; Horowitz & Hill, 2015]

### III. METODOLOGÍA

En cuanto al procedimiento seguido para la creación de los programas solución a cada problemática planteada, a continuación se detalla la metodología particular correspondiente a cada una de ellas, estructurada de forma secuencial y segmentada con el fin de facilitar la comprensión del proceso seguido.

A modo general, si bien no fue un requisito establecido en la consigna del práctico de laboratorio, se optó por desarrollar librerías específicas para cada elemento utilizado, con el objetivo de mejorar la modularidad, reducir la cantidad de líneas de código y permitir la reutilización de funciones en los distintos programas que lo requieran. Las librerías desarrolladas fueron aquellas destinadas al manejo de los módulos, periféricos, y cualquier estructura reutilizable:

PWM, UART, ADC, EEPROM, LCD, I2C, SPI, RC522 y WS2812.

Todas ellas con su respectivo archivo .h y .c.

#### A. Problema A – “Control de Plotter Electrónico”

El desarrollo de este primer sistema se basó en la evolución del trabajo realizado en los laboratorios anteriores, particularmente en el control secuencial del plotter monocromático implementado en lenguaje C durante el Laboratorio N°2.

En esta ocasión, se retomó dicha base de código y se adaptó para cumplir con los nuevos requerimientos del presente laboratorio, que contemplan el control completo de los dos motores paso a paso (ejes X e Y) junto con la válvula solenoide encargada del trazo.

En cuanto a las especificaciones del nuevo plotter, se consideraron las conexiones eléctricas provistas en la consigna oficial, resumidas en la siguiente tabla:

AVR	ATMEGA328P	ETIQUETA
PC0	PB3	CLK_X
PC2	PB4	DIR_X
PC3	PB5	EN_X
PC4	PC0	SOLENOID
PB10	PC3	CLK_Y
PB12	PC4	DIR_Y
PB14	PC5	EN_Y
PA8	PD2	LIMIT_YA
PA11	PD3	LIMIT_YD
PB6	PD5	LED

TABLE I: Asignación de pines entre AVR y ATmega328P para el control del plotter.

A partir de esta información inicial (sin contar con documentación técnica completa del equipo), fue necesario realizar pruebas experimentales para determinar el comportamiento

real de cada señal: sentido de dirección de ambos ejes, respuesta de la válvula solenoide ante niveles lógicos 1 y 0, lectura de los sensores de límite Y, entre otros parámetros.

Una vez comprendido el funcionamiento general del dispositivo, se procedió al desarrollo del código correspondiente.

Como punto de partida, se empleó el código del plotter del Laboratorio 2 a modo de base, dado que este nuevo sistema debía reproducir las mismas figuras geométricas previamente implementadas (triángulo, círculo y cruz). A partir de esa base, se realizaron las modificaciones necesarias para migrar su arquitectura a la nueva disposición de pines del plotter, manteniendo la estructura modular y la compatibilidad con las funciones existentes.

El objetivo fue preservar la mayor cantidad posible de secuencias originales, particularmente aquellas almacenadas en estructuras PROGEM, haciendo uso de la librería `javr/pgmspace.h`, indispensable para la gestión de grandes volúmenes de datos dentro de la memoria Flash. Se desarrollaron rutinas dedicadas para cada movimiento del sistema (subir, bajar, izquierda, derecha y diagonales), tanto con la solenoide activada como desactivada, garantizando continuidad y precisión en los trazos. Si bien las figuras implementadas son las mismas del laboratorio anterior, se aprovechó esta instancia para refinar los bocetos y optimizar sus secuencias.

En el caso de la cruz, se optó por rediseñarla a partir de dos diagonales en lugar de los trazos vertical y horizontal originales, logrando una representación más simétrica. Para el círculo, se reemplazó la antigua secuencia almacenada en PROGEM por una secuencia que permita la realización casi imperceptible pixel a pixel, de manera que de la impresión de ser un círculo perfecto.

#### B. Problema B – “Sistema de Control de Temperatura”

El sistema desarrollado tuvo como propósito implementar un control automático de temperatura utilizando el sensor LM35, un calefactor, un ventilador y un puente H, todo gestionado por el microcontrolador ATmega328P.

El objetivo fue mantener una temperatura dentro de un rango de confort, variando la velocidad del ventilador por PWM y activando el calefactor cuando la lectura se encontraba por debajo del punto de referencia. Como primera etapa, se configuró el módulo ADC para la lectura analógica del sensor LM35, obteniendo valores proporcionales a la temperatura ambiente.

El valor leído se convirtió a grados Celsius y se envió por UART al monitor serial, junto con la acción que realizaba el sistema en cada momento. De forma paralela, se inicializó el módulo PWM con un prescaler de 64 para el control de velocidad del ventilador y los pines de salida correspondientes al calefactor y al puente H.

El código principal se estructuró en un bucle de ejecución continua, donde en cada ciclo el sistema realizaba las siguientes operaciones:

- Lectura de la temperatura a través del ADC.

- Comparación de la lectura con los rangos definidos a partir del punto medio.
- Activación o desactivación de los actuadores según la zona detectada.

Se definieron cinco zonas de trabajo, delimitadas en torno a una variable de punto medio ajustable (inicialmente en 26 °C):

- Temperaturas bajas → calefactor encendido.
- Rango neutro → todo apagado.
- Temperaturas altas → ventilador en tres niveles de velocidad (baja, media y alta) mediante PWM.

Además, se incorporó una función interactiva por UART que permitió modificar el punto medio en tiempo real sin necesidad de reprogramar el microcontrolador.

Al presionar la tecla 'x' en la terminal, el sistema entraba en modo de pausa y solicitaba al usuario ingresar un nuevo valor de temperatura entre 10 °C y 50 °C.

Una vez recibido, el nuevo valor se validaba, se almacenaba en la variable punto medio y el sistema se anulaba la medición. Cada cambio de estado (por ejemplo, Calefactor ON, Ventilador MEDIO, etc.) se

Durante el desarrollo se verificó la correcta sincronización entre los actuadores y la comunicación serial, asegurando que las transiciones entre modos se ejecutaran de manera fluida y sin interferencias entre el PWM y el puente H.

#### C. Problema C – “Control de Motor”

El objetivo de este apartado fue diseñar un sistema que permitiera controlar la posición de un motor DC en función de la comparación entre dos potenciómetros: uno que actuaba como valor de referencia y otro acoplado al eje del motor.

El sistema debía ajustar la dirección y la velocidad del giro hasta igualar ambos valores, implementando un control proporcional simple (P). El programa comenzó configurando los dos canales del ADC para leer los valores de los potenciómetros de referencia y posición.

Luego se inicializó la comunicación UART para monitorear en tiempo real las variables de control, y se configuró el Timer0 en modo Fast PWM, utilizando el pin PD6 como salida (OC0A).

Los pines PB0 y PB1 se destinaron a la dirección del motor a través de un puente H, permitiendo invertir la polaridad según la diferencia entre ambos potenciómetros.

El control principal se basó en el cálculo del error entre la referencia (ref) y el valor actual (act).

Mediante la función MOTOR(error), el sistema determinaba la dirección de giro y calculaba el valor del PWM proporcional al módulo del error utilizando una constante de ganancia  $K_P = 0.35$ .

Se estableció además una zona muerta ( $\pm 1$ ) para evitar oscilaciones cuando el error era muy pequeño y una limitación mínima de PWM para superar la inercia del motor.

En cada ciclo del programa, se enviaban por UART los valores actuales de referencia, posición, nivel de PWM aplicado y sentido de giro (Horario, Antihorario o Detenido).

Esta información permitió visualizar la respuesta del motor y ajustar los parámetros del control proporcional de manera

precisa durante las pruebas. De esta forma, el sistema consiguió una operación estable y controlada, donde el motor respondía suavemente a los cambios del potenciómetro de referencia, alcanzando el punto deseado sin sobrepasarlo.

#### D. Problema D – “Matriz RGB con Joystick”

El desarrollo de este sistema consistió en la implementación de una matriz de LEDs WS2812 controlada mediante un joystick analógico, permitiendo mover un punto luminoso dentro de una cuadrícula 8x8 y cambiar su color presionando el botón incorporado en el mando.

El código comenzó con la inicialización de los periféricos UART, ADC y WS2812, junto con la configuración del pin PD2 como entrada con resistencia pull-up para detectar el estado del botón.

Se definió una posición inicial para el LED encendido (coordenadas 3,3) y se asignó un color aleatorio mediante la función WS2812\_COLOR\_ALEATORIO(). El joystick se utilizó para controlar la dirección del movimiento:

- Movimiento del eje X → arriba / abajo.
- Movimiento del eje Y → izquierda / derecha.
- Botón central → cambio de color aleatorio.

A través de los valores analógicos del ADC, se establecieron rangos de sensibilidad calibrados previamente para evitar movimientos bruscos o falsos.

Cada lectura del joystick actualizaba la posición del LED activo dentro de la matriz y enviaba la información por UART, mostrando las coordenadas, dirección detectada y color actual (R,G,B).

El ciclo principal realizaba la secuencia de limpieza, actualización de color y visualización en la matriz utilizando la librería ws2812.c, que gestiona la comunicación de un solo hilo necesaria para estos LEDs. Un retardo de 150 ms entre ciclos permitió una transición suave y una lectura estable del joystick.

El resultado fue un sistema interactivo y visualmente atractivo, donde el usuario podía mover el punto luminoso con precisión y cambiar su color dinámicamente desde el joystick, combinando entradas analógicas y salidas digitales en una práctica integradora.

#### E. Problema E – “Cerradura RFID”

El desarrollo de este sistema tuvo como objetivo implementar una cerradura electrónica controlada por el microcontrolador ATmega328P, utilizando un lector RFID RC522 para la lectura y validación de tarjetas.

A diferencia de prácticas anteriores, en esta ocasión se integraron varios periféricos de forma simultánea, combinando SPI, I2C, UART y memoria EEPROM para lograr un funcionamiento completo y autónomo del sistema.

El funcionamiento general se basa en tres modos principales: detección, registro y borrado, los cuales determinan la tarea que el sistema realiza en cada momento.

Durante el modo de detección, el lector RFID busca una tarjeta cercana y, al encontrarla, compara su ID con el valor guardado en la EEPROM. Si los identificadores coinciden, se

enciende el LED verde y se muestra en la pantalla LCD el mensaje “Acceso permitido”, acompañado de un breve beep de confirmación. En caso contrario, se activa el LED rojo, el sistema emite dos beeps de advertencia y en pantalla se indica “Acceso denegado”.

Todos los eventos se envían también por puerto UART, permitiendo supervisar desde la terminal serial las lecturas y respuestas del sistema. El modo de registro se utiliza para guardar una nueva tarjeta como clave de acceso.

Al presionar el botón correspondiente, la LCD muestra “Registrar – acerque tarjeta” y el sistema queda a la espera de un ID válido. Una vez detectado, el número de identificación se guarda de forma permanente en la EEPROM, confirmando la acción mediante un beep único y el mensaje “Nueva tarjeta registrada”.

El modo de borrado, por su parte, permite eliminar la tarjeta guardada. Al presionar el botón asignado, la memoria se limpia, se muestra “Tarjeta borrada” y el sistema vuelve al estado inicial de detección, listo para registrar o verificar una nueva tarjeta.

Entendiendo el comportamiento y el diagrama de estados planteado, se procedió a comenzar con el desarrollo del código. Para ello, se estructuró el programa de forma modular, utilizando librerías propias para cada periférico involucrado. Las primeras rutinas implementadas fueron las de inicialización de los protocolos de comunicación y de los pines de entrada/salida.

Una vez finalizada esta etapa, el sistema mostró en pantalla el mensaje “Bienvenido al sistema RFID”, indicando el correcto inicio del programa. Posteriormente, se desarrolló la lógica principal mediante una estructura switch-case que gestionó los tres modos de operación del sistema. Cada rutina fue validada por separado y luego integrada al programa completo.

Durante las pruebas, tanto en simulación como en montaje físico, se comprobó el correcto funcionamiento de los modos y la interacción entre los módulos de hardware, verificando la coherencia de los mensajes en LCD, buzzer, LEDs y comunicación serial.

#### IV. RESULTADOS

A continuación, se presentan los resultados obtenidos para cada una de las problemáticas planteadas, luego de haber seguido la metodología específica correspondiente a cada caso. La información se segmenta por problema para mantener una estructura clara y coherente.

##### A. Problema A – “Control de Plotter Electrónico”

En el caso del plotter electrónico, se lograron establecer con éxito todas las funciones y rutinas necesarias para su correcto funcionamiento, permitiendo una reproducción precisa de los patrones y secuencias de trazos de las figuras solicitadas.

Durante el desarrollo fue necesario analizar cuidadosamente el comportamiento de cada movimiento, especialmente en lo referente al control de los ejes.

Por ejemplo, para los desplazamientos en el eje X (derecha e izquierda), se determinó que era necesario aplicar un nivel lógico alto (5 V) al pin EN\_X (Enable) y a la línea de dirección correspondiente, según el sentido del movimiento.

Se implementaron dos tipos de secuencias:

- Secuencias de trazo, donde la solenoide permanece baja, permitiendo dibujar.
- Secuencias de movimiento libre, con la solenoide levantada, para reposicionar el cabezal sin marcar el papel.

El punto más relevante del desarrollo fue la generación de los pulsos de paso (CLK) para el control de los motores paso a paso en ambos ejes.

Inicialmente se evaluó la posibilidad de utilizar un PWM por hardware en los pines de control, pero se identificó una limitación: el pin PC3 (CLK del eje Y) no cuenta con salida PWM nativa, mientras que el del eje X sí. Para mantener uniformidad en el comportamiento, se optó finalmente por generar los pulsos por software en ambos ejes, asegurando una sincronización homogénea y simplificando el control de velocidad y dirección.

Estos pulsos se aplicaron en cada rutina de movimiento, generando desplazamientos individuales sobre cada eje o combinados en movimientos diagonales, donde ambos motores actuaron de forma simultánea para lograr los trazos oblicuos requeridos.

Tal como se mencionó en la metodología, se descartó la comunicación UART para el envío de comandos debido a que el plotter no contaba con la configuración ni la compatibilidad necesarias para ese tipo de control.

Por ello, las secuencias fueron ejecutadas directamente desde la función principal, adaptando la rutina FIGURAS() para incorporar las cinco formas requeridas —triángulo, cruz, círculo, zorro y flor— dentro de una única ejecución. En cuanto a las figuras, se mantuvieron las originales, pero se realizaron leves modificaciones para optimizar su trazado:

- En la cruz, se reemplazaron los trazos verticales y horizontales por movimientos diagonales de 45°, logrando una figura más simétrica y estéticamente definida.
- En el círculo, se abandonó la estructura basada en secuencias programadas (PROGMEM) y se diseñó una nueva función que utiliza coordenadas polares, generando microtrazos lineales que, a simple vista, conforman un círculo continuo con excelente resolución.

Finalmente, para las figuras del zorro geométrico y la flor, se realizaron ajustes menores en las secuencias de bajada de la solenoide, ya que en el código anterior esta acción estaba integrada dentro de las funciones de movimiento, mientras que en esta nueva versión se controla de manera independiente. Además, ambas figuras fueron reescaladas y adaptadas para trazarse junto con las figuras básicas en una misma hoja, optimizando el área de trabajo del plotter y manteniendo la coherencia del conjunto.

### *B. Problema B – “Sistema de Control de Temperatura”*

En este problema se desarrolló un sistema capaz de regular la temperatura ambiente utilizando un calefactor y un ventilador controlados por PWM, en función de la lectura del sensor LM35. El sistema mantuvo la temperatura dentro de un rango definido a partir de una variable denominada punto medio, conmutando entre calefacción y ventilación según la lectura obtenida.

Durante las pruebas, el programa respondió correctamente en cada una de las zonas de operación. Cuando la temperatura se encontraba por debajo del punto medio, el calefactor se encendía hasta alcanzar el rango neutro; dentro de ese rango, todos los actuadores permanecían apagados; y por encima del punto medio, el ventilador se activaba en tres niveles de velocidad (bajo, medio y alto) según la diferencia detectada.

El sistema incluyó la posibilidad de ajustar el punto medio en tiempo real mediante la comunicación UART, ingresando un nuevo valor entre 10 °C y 50 °C sin necesidad de reprogramar el microcontrolador. Esta función se validó con éxito, observando que el sistema actualizaba los límites de trabajo y reanudaba la lectura de temperatura automáticamente.

Los mensajes enviados por UART mostraron los valores actuales de temperatura, el punto medio configurado y la acción en curso, confirmando el correcto funcionamiento del control y la estabilidad de las transiciones. En conjunto, el sistema mantuvo una respuesta estable y coherente frente a los cambios de temperatura simulados y reales.

En cuanto a la actualización del punto medio, hubo problemas con el funcionamiento del mismo, ya que las lecturas del valor ADC fluctuaban cuando se colocaba un valor del PM de más de 30 grados.

### *C. Problema C - “Control de Motor”*

En este problema se implementó un sistema de control de posición de un motor DC utilizando dos potenciómetros: uno de referencia y otro acoplado al eje del motor. El microcontrolador comparó ambos valores y ajustó la dirección y la velocidad del motor hasta igualarlos, aplicando un control proporcional simple.

Durante el funcionamiento, las lecturas de ambos potenciómetros fueron estables y el motor respondió de manera progresiva a los cambios en la referencia. Cuando el valor del potenciómetro de referencia era mayor que el actual, el motor giraba en sentido horario; cuando era menor, lo hacía en sentido antihorario; y al alcanzar la posición deseada, se detenía completamente dentro de la zona muerta definida.

La señal PWM aplicada al motor varió correctamente según la magnitud del error, aumentando la velocidad a medida que la diferencia entre ambos potenciómetros era mayor. Las pruebas mostraron una respuesta fluida y sin oscilaciones, con un control de dirección preciso y sin saltos de tensión en el puente H.

Los datos enviados por UART reflejaron en todo momento las lecturas de referencia y posición, el valor del PWM aplicado y el sentido de giro, permitiendo verificar el comportamiento del sistema en tiempo real. En general,

el control de motor cumplió con los objetivos propuestos, logrando un movimiento estable, con respuesta proporcional y funcionamiento continuo.

### *D. Problema D - “Matriz RGB con Joystick”*

Para este problema, se realizó con éxito la implementación de la matriz RGB controlada por el integrado WS2812, junto con el manejo de un joystick analógico que permitió mover un LED activo, ubicado inicialmente en el centro de la matriz, a través de toda su superficie. El sistema respetó los límites físicos de la matriz y, además, permitió cambiar el color del LED de forma aleatoria al presionar el botón del joystick.

Para su ejecución, se hizo uso de la librería creada personalmente para el control y manejo del controlador WS2812, encargada de gestionar la transmisión de datos a los LEDs mediante un único hilo de comunicación, manteniendo la temporización exacta requerida por el protocolo.

También se emplearon las librerías personalizadas para el ADC, utilizado en la lectura de los ejes X e Y del joystick, y para la UART, que permitió visualizar por consola las posiciones y valores leídos en cada momento. El programa se estructuró en un bucle continuo donde, en cada ciclo, se leen los valores analógicos de ambos ejes para determinar la dirección del movimiento.

Se establecieron rangos de sensibilidad calibrados para evitar lecturas falsas y definir claramente cuándo el joystick se desplazaba hacia arriba, abajo, izquierda o derecha. De esta manera, el LED se mueve dentro de la matriz paso a paso, sin salirse de los bordes.

Cuando se detecta la pulsación del botón del joystick, el sistema genera un nuevo color aleatorio mediante la función correspondiente, actualizando el LED activo.

Cada acción se muestra también por UART, indicando las coordenadas, la dirección detectada y los valores RGB actuales. Durante las pruebas, la matriz respondió correctamente a cada desplazamiento, mostrando un movimiento fluido y preciso. El cambio de color se realizó sin errores y el retardo entre lecturas permitió una actualización estable.

En conjunto, el sistema funcionó como se esperaba, permitiendo un control visual y dinámico del LED dentro de la matriz, combinando entradas analógicas, comunicación digital y retroalimentación visual en un mismo proyecto.

### *E. Problema E - “Cerradura RFID”*

En este problema se logró la implementación completa y estable de la cerradura electrónica con identificación mediante tarjetas RFID, controlada por el ATmega328P y el lector RC522.

El sistema funcionó correctamente en sus tres modos de operación: detección, registro y borrado, mostrando un comportamiento estable y predecible en todas las pruebas realizadas.

Durante las pruebas se verificó que la lectura del módulo RC522 fue constante y sin errores, detectando las tarjetas a una distancia adecuada y con tiempos de respuesta rápidos.

En el modo de detección, al acercar una tarjeta registrada, el LED verde se encendió de inmediato y el buzzer emitió

un beep corto, mientras que ante una tarjeta no registrada el sistema encendió el LED rojo, emitió dos beeps y mostró “Acceso denegado” en la pantalla LCD.

El modo registro funcionó correctamente en todas las pruebas, guardando nuevas tarjetas en la EEPROM sin fallas y manteniendo la información almacenada incluso tras cortar la alimentación.

El modo borrado eliminó el contenido de memoria y reinició el sistema al estado de detección, mostrando en pantalla el mensaje “Tarjeta borrada” y confirmando la operación por UART.

Los mensajes en la LCD, el buzzer y los LEDs indicadores se mantuvieron sincronizados en todo momento con las acciones ejecutadas, facilitando la verificación visual del funcionamiento.

La comunicación por UART permitió observar las lecturas de las tarjetas y los mensajes enviados por el sistema en tiempo real.

En general, el sistema se comportó de forma estable tanto en simulación como en montaje físico. Los modos de operación respondieron sin interferencias entre sí y los periféricos se comunicaron correctamente.

## V. CONCLUSIÓN

El presente trabajo permitió integrar en un único proceso experimental diversos principios de programación embebida, adquisición de datos y control electrónico, aplicados sobre el microcontrolador ATmega328P. A lo largo de las distintas problemáticas abordadas se consolidaron conocimientos en manejo de periféricos, desarrollo modular de firmware y sincronización entre hardware y software, validando el comportamiento de los sistemas tanto en simulación como en montaje físico.

En el problema A, se alcanzó un control completo y estable del trazador gráfico mediante la generación de pulsos por software y la gestión independiente de los ejes X e Y. La implementación de secuencias optimizadas permitió obtener figuras precisas y trazos continuos, reafirmando la importancia del temporizado exacto y la coordinación de movimientos en aplicaciones de control de posición.

En el caso del problema B, el sistema logró mantener la temperatura dentro de un rango definido mediante el uso combinado del sensor LM35, el calefactor y el ventilador regulado por PWM. La posibilidad de ajustar el punto medio por UART evidenció una comunicación bidireccional eficaz y una respuesta estable frente a variaciones térmicas, aplicando de forma práctica un control automático de lazo cerrado.

En cuanto al C, se verificó el correcto funcionamiento de un control proporcional simple (P) basado en la comparación de dos señales analógicas. El motor respondió de forma progresiva y sin oscilaciones notorias, confirmando la eficiencia del método para lograr una realimentación continua y estable. La incorporación del monitoreo serial permitió

cuantificar con precisión la relación entre error, dirección y velocidad de respuesta.

A su vez, en el problema D, la matriz de LEDs WS2812 respondió con exactitud al desplazamiento del joystick, demostrando la sincronización adecuada entre la lectura analógica y la actualización digital del color. La interacción entre usuario y sistema fue fluida y confiable, validando el uso de controladores RGB direccionables como una herramienta efectiva para interfaces visuales dinámicas.

Por último, en el problema E, el sistema de identificación por radiofrecuencia funcionó de manera robusta en sus tres modos de operación: detección, registro y borrado. La integración de EEPROM, LCD I<sup>2</sup>C, buzzer y comunicación SPI con el lector RC522 consolidó una aplicación embebida compleja, destacando la interoperabilidad entre múltiples periféricos y la persistencia de datos no volátil.

En conjunto, las actividades de laboratorio confirmaron la capacidad del ATmega328P para gestionar simultáneamente distintos protocolos de comunicación y técnicas de control. La experiencia adquirida fortaleció competencias en diseño modular, depuración estructurada y validación experimental, sentando las bases para el desarrollo de sistemas mecatrónicos integrados, confiables y escalables en futuros proyectos.

## REFERENCES

- [1] Microchip Technology Inc., *ATmega328P – 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*, Chandler, AZ, USA: Microchip Technology Inc., 2020. [Online]. Available: <https://www.microchip.com/en-us/product/ATmega328P>
- [2] Arduino AG, *Arduino UNO Rev3 Technical Specifications*, Ivrea, Italy: Arduino AG, 2024. [Online]. Available: <https://docs.arduino.cc/hardware/uno-rev3>
- [3] IEEE, *IEEE Editorial Style Manual*, Piscataway, NJ, USA: IEEE, 2023. [Online]. Available: <https://journals.ieeeauthorcenter.ieee.org>
- [4] Texas Instruments, *LM35 Precision Centigrade Temperature Sensors*, Dallas, TX, USA: Texas Instruments, 2018. [Online]. Available: <https://www.ti.com/lit/ds/symlink/lm35.pdf>
- [5] Adafruit Industries, *WS2812 Intelligent Control LED Datasheet*, New York, NY, USA: Adafruit Industries, 2019. [Online]. Available: <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>
- [6] NXP Semiconductors, *MFRC522 – RFID Reader IC Datasheet*, Eindhoven, Netherlands: NXP Semiconductors, 2019. [Online]. Available: <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>
- [7] Motorola Inc., *SPI Block Guide*, Phoenix, AZ, USA: Motorola Semiconductor Products, 1985.
- [8] NXP Semiconductors, *I<sup>2</sup>C-bus Specification and User Manual*, Rev. 6, Eindhoven, Netherlands: NXP, 2014. [Online]. Available: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- [9] R. W. Erickson and D. Maksimović, *Fundamentals of Power Electronics*, 2nd ed., Boston, MA, USA: Kluwer Academic Publishers, 2001.
- [10] M. Pont, *Embedded C*, 2nd ed., Boston, MA, USA: Addison-Wesley, 2002.
- [11] P. Horowitz and W. Hill, *The Art of Electronics*, 3rd ed., Cambridge, U.K.: Cambridge University Press, 2015.
- [12] A. Hughes and B. Drury, *Electric Motors and Drives: Fundamentals, Types and Applications*, 5th ed., Oxford, U.K.: Butterworth-Heinemann, 2019.
- [13] R. L. Boylestad and L. Nashelsky, *Electronic Devices and Circuit Theory*, 12th ed., Upper Saddle River, NJ, USA: Pearson, 2019.
- [14] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 8th ed., New York, NY, USA: Oxford University Press, 2016.



- [15] Microchip Technology Inc., *Microchip Studio for AVR and SAM Devices – Integrated Development Environment*, Chandler, AZ, USA: Microchip Technology Inc., 2023. [Online]. Available: <https://www.microchip.com/en-us/tools-resources/develop/microchip-studio>

## VI. APÉNDICE

### *Apéndice A – Repositorio GitHub*

Con el fin de garantizar la trazabilidad y disponibilidad de los códigos desarrollados en este laboratorio, se utilizó un repositorio en GitHub como plataforma de control de versiones y colaboración. En él se documentan los programas en lenguaje C implementados para cada problemática, así como las evidencias complementarias (fotos, videos y diagramas) y las simulaciones correspondientes.

El repositorio está disponible en el siguiente enlace:

### **REPOSITORIO**