

# Laboratorio 3

1<sup>st</sup> Luis David Nuñez Noguera  
*Universidad Tecnológica del Uruguay ITRSO*  
Fray Bentos, Uruguay  
luis.nunez@estudiantes.utec.edu.uy

2<sup>nd</sup> Tiago Corrales Martinez  
*Universidad Tecnológica del Uruguay ITRSO*  
Fray Bentos, Uruguay  
tiago.corrales@estudiantes.utec.edu.uy

**Index Terms**—Plotter, PWM, Sensor temperatura, Cerradura electrónica

**Abstract**—El presente trabajo describe el desarrollo e implementación de distintos sistemas de control basados en el microcontrolador ATmega328P. Se intentaron llevar a cabo cinco prácticas principales: el control de un plotter electrónico, un sistema de regulación de temperatura (El cual no fue realizado), el control de un motor con referencia analógica, una matriz RGB interactiva y una cerradura electrónica RFID.

Cada proyecto integró sensores, actuadores y distintos métodos de comunicación serial, permitiendo aplicar de manera práctica los conceptos de electrónica digital, control automático y programación en lenguaje C. Los resultados obtenidos demostraron la correcta operación de los sistemas diseñados tanto en simulación como en implementación física, validando el funcionamiento del hardware y del software desarrollado.

El trabajo permitió afianzar competencias en diseño de sistemas embebidos, manejo de periféricos y análisis de señales analógicas y digitales, resaltando la importancia del enfoque experimental en el proceso de aprendizaje y en la consolidación de habilidades técnicas aplicadas.

## I. INTRODUCCIÓN

El presente informe corresponde al Laboratorio 3 de la Unidad Curricular *Tecnologías de Microprocesamiento*, del cuarto semestre de la carrera de Ingeniería Mecatrónica. El trabajo tiene como propósito la implementación y análisis de distintos sistemas basados en el microcontrolador ATmega328P, integrando el uso de sensores, actuadores y técnicas de comunicación serial para el control de dispositivos electrónicos.

El laboratorio se divide en cinco partes principales, cada una orientada al desarrollo de una aplicación distinta: control de un plotter electrónico, sistema de control de temperatura, control de motor con referencia analógica, matriz RGB controlada por joystick y una cerradura electrónica RFID. Estas experiencias permiten aplicar de forma práctica los conocimientos de electrónica digital, programación en lenguaje C y control automático, reforzando la comprensión del funcionamiento de los sistemas embebidos y de seguridad electrónica.

El enfoque de este trabajo busca no solo la correcta implementación física y simulada de los sistemas, sino también la documentación estructurada del proceso, respetando el formato de redacción IEEE. Con ello, se promueve la capacidad de comunicar resultados técnicos de forma clara, precisa y profesional.

## II. OBJETIVOS

### OBJETIVO GENERAL

Desarrollar y evaluar distintos sistemas de control basados en el microcontrolador ATmega328P, aplicados a escenarios prácticos como el control de un plotter electrónico, un sistema de regulación de temperatura, un control de motor con referencia analógica, una matriz RGB interactiva y una cerradura electrónica RFID, integrando sensores, actuadores y algoritmos de control con el fin de fortalecer las competencias en programación, diseño y análisis de sistemas embebidos en un entorno de laboratorio.

### OBJETIVOS ESPECÍFICOS

- Implementar sistemas de control de temperatura, posición y visualización RGB utilizando sensores, actuadores y módulos de comunicación, evaluando su comportamiento en simulación y pruebas físicas.
- Desarrollar interfaces de interacción con el usuario mediante periféricos como joysticks, pantallas y módulos de entrada analógica, optimizando la respuesta del sistema ante diferentes condiciones de operación.
- Validar el comportamiento de todos los sistemas tanto en simulación como en práctica de laboratorio, analizando el desempeño, precisión y robustez de cada implementación.

## III. MATERIALES

En el desarrollo de las diferentes etapas del Laboratorio 3 se emplearon múltiples componentes electrónicos, dispositivos de control y elementos de conexión. A continuación, se detallan los materiales utilizados en cada una de las partes experimentales.

### A. Parte A – Control de Plotter Electrónico

- **Microcontrolador ATmega328P:** Unidad central encargada de controlar los movimientos de los motores paso a paso, la activación del solenoide y la comunicación serial mediante UART.
- **Motores paso a paso:** Dos unidades utilizadas para generar los desplazamientos en los ejes X e Y del plotter.
- **Solenoide electromecánico:** Mecanismo encargado de levantar o descender el lápiz de trazo.
- **Sensores de límite:** Cuatro sensores ubicados en los extremos de los ejes para delimitar el área de dibujo, aunque solo dos funcionales.
- **Fuente de alimentación 5 V / 12 V.**

- **Interfaz UART–USB:** Comunicación con el computador para envío de órdenes.
- **Cableado y protoboard.**

#### B. Parte B – Sistema de Control de Temperatura

- **Microcontrolador ATmega328P.**
- **Sensor de temperatura LM35.**
- **Calefactor resistivo y ventilador DC.**
- **Transistores de potencia y resistencias de control.**
- **Interfaz UART** para monitoreo de datos.
- **Fuente de 5 V DC.**

#### C. Parte C – Control de Motor con Referencia Analógica

- **Microcontrolador ATmega328P.**
- **Motor DC.**
- **Potenciómetro de referencia y potenciómetro en el motor.**
- **Driver L293D.**
- **Fuente de 5 V / 12 V DC.**
- **Interfaz UART.**

#### D. Parte D – Matriz RGB con Joystick

- **Microcontrolador ATmega328P.**
- **Matriz de LEDs RGB 8x8 WS2812.**
- **Joystick analógico con pulsador.**
- **Fuente de 5 V DC.**
- **Cableado y protoboard.**

#### E. Parte E – Cerradura Electrónica RFID

- **Microcontrolador ATmega328P:** Procesa la identificación de las tarjetas RFID y controla el mecanismo de bloqueo.
- **Módulo RFID RC522:** Permite la lectura de tarjetas y llaveros mediante tecnología de radiofrecuencia.
- **Buzzer y LED indicadores:** Proveen retroalimentación auditiva y visual sobre el estado de acceso (concedido o denegado).
- **Display LCD 16x2 I2C:** Muestra mensajes de bienvenida, error y autenticación.
- **Fuente de 5 V DC.**
- **Cableado y protoboard.**

### IV. MARCO TEÓRICO

Antes de detallar el procedimiento llevado a cabo, es necesario clarificar distintos conceptos importantes para la comprensión total de la práctica.

#### A. ATmega328p

Un microcontrolador puede definirse como un circuito integrado que incorpora en un solo dispositivo una unidad central de procesamiento (CPU), memoria volátil y no volátil, así como periféricos de entrada y salida programables. A diferencia de los microprocesadores de propósito general, los microcontroladores están diseñados para ejecutar tareas específicas de control en sistemas electrónicos.

Este tipo de dispositivos constituye el núcleo de los sistemas embebidos, los cuales se caracterizan por estar dedicados a

una función concreta dentro de un sistema mayor. Dichos sistemas se encuentran presentes en una amplia variedad de aplicaciones cotidianas y de carácter industrial, tales como electrodomésticos, automóviles, dispositivos médicos, equipos de telecomunicaciones y sistemas de automatización. Su relevancia radica en que permiten el diseño de soluciones con bajo consumo energético, tamaño reducido, costo accesible y alta confiabilidad [1].

1) *Características básicas del ATmega328P:* El ATmega328P, fabricado por Microchip Technology (anteriormente Atmel), es uno de los microcontroladores más representativos de la arquitectura AVR de 8 bits. Es ampliamente conocido por ser el componente central de placas de desarrollo como la Arduino Uno, lo que lo ha convertido en un referente tanto en entornos educativos como en aplicaciones prototípicas y comerciales [2].

Entre sus principales características técnicas destacan:

- Arquitectura AVR RISC de 8 bits, que permite ejecutar la mayoría de instrucciones en un solo ciclo de reloj y optimizar el rendimiento mediante un pipeline sencillo.
- Memoria integrada, compuesta por 32 KB de memoria Flash programable, 2 KB de SRAM y 1 KB de EEPROM, lo que posibilita almacenar tanto el programa como datos temporales y permanentes.
- Puertos de entrada y salida digital (GPIO), con un total de 23 líneas configurables, capaces de manejar señales de entrada o salida y de activar resistencias pull-up internas.
- Módulos temporizadores y contadores, que incluyen dos de 8 bits y uno de 16 bits, con soporte para funciones de modulación por ancho de pulso (PWM).
- Interfaces de comunicación serial, como USART (síncrona y asíncrona), SPI (Serial Peripheral Interface) y TWI (I<sup>2</sup>C), lo que facilita la conexión con otros microcontroladores o periféricos externos.
- Conversor analógico–digital (ADC) de 10 bits con hasta seis canales, que posibilita la digitalización de señales analógicas.
- Capacidad de operación hasta los 20 MHz y disponibilidad de múltiples modos de bajo consumo, lo cual lo hace adecuado para aplicaciones portátiles y autónomas.

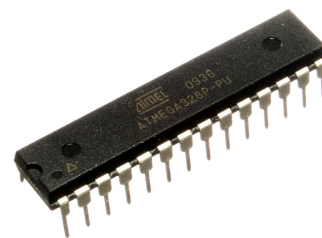


Fig. 1. Microcontrolador ATMEGA328P

## B. Comunicación Serial (USART)

La comunicación serial es un método de transmisión de datos en el cual la información se envía bit a bit a través de un canal de comunicación. Dentro de esta categoría, la interfaz UART (Universal Asynchronous Receiver and Transmitter) y su extensión USART (Universal Synchronous and Asynchronous Receiver and Transmitter) son ampliamente utilizadas en microcontroladores y sistemas embebidos [3].

En el modo asíncrono, la transmisión no requiere una señal de reloj compartida entre emisor y receptor. En su lugar, ambos dispositivos deben acordar previamente parámetros como la velocidad de transmisión (baud rate) y el formato de la trama de datos. La sincronización se logra mediante bits especiales que indican el inicio y fin de cada paquete transmitido.

Por otro lado, en el modo síncrono (propio de USART), se utiliza una línea de reloj adicional, lo cual permite mayor velocidad y confiabilidad en la comunicación. Sin embargo, el modo asíncrono UART sigue siendo el más utilizado debido a su simplicidad y menor requerimiento de pines.

1) *Formato de transmisión*: La transmisión de datos mediante UART/USART sigue un formato estructurado de trama, compuesto por los siguientes elementos:

- **Bit de inicio (Start bit)**: Indica el comienzo de la transmisión; normalmente consiste en un nivel lógico bajo (0).
- **Bits de datos**: Generalmente entre 5 y 9 bits, aunque lo más común es el uso de 8 bits, lo que permite la transmisión de un byte por trama.
- **Bit de paridad (opcional)**: Permite la detección básica de errores en la transmisión, pudiendo configurarse como par, impar o sin paridad.
- **Bit(s) de parada (Stop bit)**: Señalizan el final de la trama; pueden configurarse en 1 o 2 bits.
- **Velocidad de transmisión (baud rate)**: Corresponde al número de símbolos transmitidos por segundo. En UART, un baud equivale a un bit, por lo que baud rate y bit rate son equivalentes. Valores comunes son 9600, 19200 y 115200 baudios, dependiendo de la aplicación.

La correcta configuración de estos parámetros en ambos extremos es fundamental para garantizar una comunicación confiable y libre de errores.



Fig. 2. Comunicación UART

## C. Plotter

Un plotter es un dispositivo de impresión controlado por computadora que permite la representación gráfica de datos mediante trazos lineales. Se utiliza ampliamente en campos

como la ingeniería, el diseño asistido por computadora (CAD) y la arquitectura [4]. El principio de funcionamiento se basa en el movimiento coordinado de ejes motorizados que posicionan un cabezal de dibujo sobre una superficie.

En el contexto del laboratorio, el plotter se controla mediante un Controlador Lógico Programable (PLC), encargado de gestionar los actuadores responsables del movimiento y del control neumático del trazo. El sistema cuenta con mecanismos de seguridad, incluyendo un pulsador de emergencia (E-STOP) y un botón de reinicio (RESET), que restablece la posición inicial del cabezal. La Tabla I muestra la asignación de pines utilizada para la comunicación entre el microcontrolador y el PLC.

TABLE I  
PINES PARA MANEJO DE PLOTTER

Módulo AVR	ATmega328P	Etiqueta
PC0	PD3	CLK X
PC2	PD4	DIR X
PC3	PD5	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

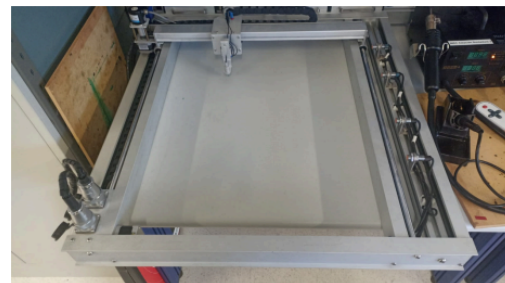


Fig. 3. Plotter

## D. Pantalla LCD 16x2 con interfaz I<sup>2</sup>C

Una pantalla LCD (Liquid Crystal Display) de 16x2 es un dispositivo de salida utilizado comúnmente en sistemas embebidos para la visualización de información alfanumérica. El número "16x2" indica que el módulo posee dos líneas capaces de mostrar hasta dieciséis caracteres cada una. Este tipo de pantallas se basa en la tecnología de cristal líquido, la cual controla la orientación de moléculas entre dos capas conductoras para modular la cantidad de luz transmitida y así formar los caracteres visibles.

El módulo LCD 16x2 requiere tradicionalmente múltiples conexiones paralelas para su control (generalmente 8 líneas de datos y varias de control). Sin embargo, mediante el uso de un expansor de bus I2C (como el PCF8574), es posible reducir significativamente el número de pines necesarios para su conexión, pasando de un esquema paralelo a una comunicación serial de dos líneas: SDA (Serial Data) y SCL (Serial Clock).

La interfaz I2C (Inter-Integrated Circuit) permite conectar múltiples dispositivos esclavos utilizando solo estas dos líneas, cada uno con una dirección única. En el caso del LCD 16x2, el módulo I2C actúa como un puente que traduce las señales seriales en comandos de control para el controlador interno del display (habitualmente el HD44780 o compatible). Esto simplifica el cableado, reduce el consumo de pines del microcontrolador y facilita la integración en aplicaciones de automatización o monitoreo.

El control del LCD incluye comandos para posicionar el cursor, limpiar la pantalla, activar o desactivar el retroiluminado, y escribir caracteres personalizados en la memoria CGRAM. Estos comandos pueden enviarse mediante bibliotecas específicas para el microcontrolador, como la librería `LiquidCrystal_I2C` en entornos de desarrollo Arduino, lo que permite una programación más sencilla y estructurada [5].



Fig. 4. Pantalla LCD con interfaz I2C

#### E. Memoria EEPROM

La memoria EEPROM (Electrically Erasable Programmable Read-Only Memory) es un tipo de memoria no volátil que permite conservar los datos almacenados incluso cuando el dispositivo pierde la alimentación eléctrica. A diferencia de la memoria Flash o la ROM tradicional, la EEPROM puede ser reprogramada eléctricamente a nivel de bytes, lo que la hace especialmente útil para guardar configuraciones, parámetros de calibración o registros que deben mantenerse entre encendidos.

En los microcontroladores, como el ATmega328P, la EEPROM suele estar integrada en el propio chip, proporcionando una capacidad de almacenamiento típica del orden de 512 a 1024 bytes. Su acceso se realiza mediante instrucciones específicas o funciones de biblioteca que permiten leer o escribir datos de manera controlada, evitando la corrupción de memoria durante el proceso de escritura.

El tiempo de escritura de la EEPROM es mayor que el de la memoria RAM o Flash, debido a la naturaleza de su tecnología de almacenamiento, que implica procesos de carga eléctrica en celdas capacitivas. Sin embargo, su principal ventaja radica en su durabilidad, permitiendo hasta 100,000 ciclos de escritura/borrado por celda, lo cual resulta suficiente para la mayoría de aplicaciones embebidas [6].

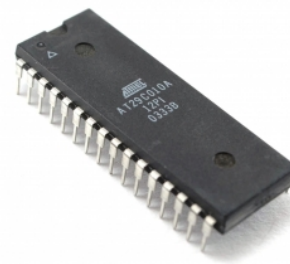


Fig. 5. Memoria EEPROM

#### F. Sensor LM35

El LM35 es un sensor de temperatura analógico lineal fabricado por Texas Instruments, diseñado para proporcionar una salida de voltaje proporcional a la temperatura en grados Celsius. A diferencia de los sensores de tipo resistivo o termopares, el LM35 no requiere calibración externa y ofrece una precisión típica de  $\pm 0.5^\circ\text{C}$  a temperatura ambiente [?].

La relación entre la temperatura y la tensión de salida es directa, siendo de  $10\text{ mV}/^\circ\text{C}$ . Esto implica que, por ejemplo, a una temperatura de  $25^\circ\text{C}$  el sensor entrega una tensión de 250 mV. El rango de medición típico se extiende entre  $-55^\circ\text{C}$  y  $+150^\circ\text{C}$ , lo que lo hace adecuado para aplicaciones industriales, ambientales y de control de procesos.

La señal de salida del LM35 puede conectarse directamente a una entrada analógica del microcontrolador, donde el convertor analógico-digital (ADC) convierte el voltaje en una lectura digital. A partir de esta lectura, es posible calcular la temperatura real mediante una simple relación de proporcionalidad. Además, su baja impedancia de salida y consumo reducido permiten integrarlo fácilmente en sistemas de monitoreo o regulación térmica, como control de ventiladores o calefactores.

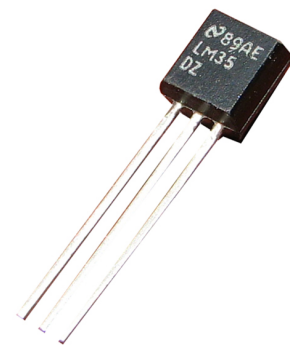


Fig. 6. Sensor de temperatura LM35

#### G. Generación de Señal PWM

La modulación por ancho de pulso (PWM, por sus siglas en inglés) es una técnica utilizada para controlar la cantidad de energía entregada a una carga mediante la variación del ciclo de trabajo de una señal periódica digital. En un ciclo de trabajo (*duty cycle*) del 0% la señal permanece siempre en nivel bajo,

mientras que en un 100% se mantiene permanentemente en nivel alto.

En microcontroladores como el ATmega328P, la generación de señales PWM se realiza mediante los módulos de temporizadores internos. Estos comparan un valor de referencia almacenado en un registro (OCRx) con el contador del temporizador, determinando el momento en que la salida cambia de estado. La frecuencia de la señal PWM depende del reloj del sistema y de los valores de preescalado configurados en el temporizador.

El PWM tiene múltiples aplicaciones en electrónica de control: regulación de velocidad de motores de corriente continua (DC), control de brillo en LEDs, conversión de señales analógicas simuladas y control de servomotores. Su eficiencia radica en que la conmutación digital reduce las pérdidas por disipación, a diferencia del control lineal [9].

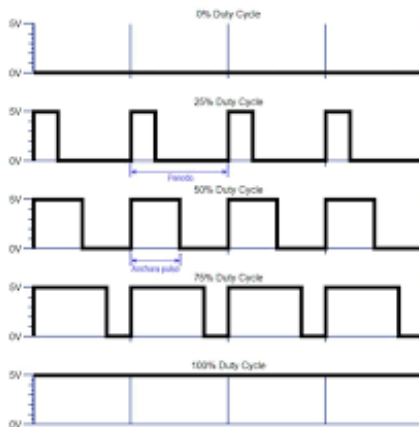


Fig. 7. Representación gráfica de la modulación PWM con distintos ciclos de trabajo

#### H. Tira de LEDs WS2812

La tira de LEDs WS2812 está compuesta por diodos emisores de luz RGB controlables individualmente mediante un único bus de datos digital. Cada módulo WS2812 integra un chip controlador que interpreta la señal y ajusta la intensidad de cada componente de color (rojo, verde y azul) mediante modulación por ancho de pulso (PWM). El protocolo de comunicación es de tipo unidireccional y de tiempo crítico, requiriendo precisión en la generación de los pulsos para codificar correctamente los valores de color de 24 bits por LED. Gracias a su facilidad de uso y a la posibilidad de controlar múltiples LEDs con un solo pin, se emplea ampliamente en proyectos de iluminación decorativa, interfaces visuales y sistemas embebidos interactivos [7].



Fig. 8. Matriz LED

#### I. Módulo joystick

Un joystick es un dispositivo de entrada analógico utilizado para capturar movimientos bidimensionales. Generalmente está constituido por dos potenciómetros dispuestos ortogonalmente que permiten medir desplazamientos en los ejes X e Y. Cada potenciómetro actúa como un divisor resistivo, entregando un voltaje proporcional a la posición del eje correspondiente.

Los valores de voltaje de salida del joystick suelen variar entre 0 V y la tensión de alimentación (por ejemplo, 5 V), con una posición central que representa el valor medio (aproximadamente 2.5 V). Estas señales analógicas se aplican a las entradas ADC del microcontrolador, donde son digitalizadas para su posterior procesamiento. En algunos modelos, el joystick incorpora además un pulsador integrado que permite la detección de eventos de selección o confirmación.

El uso del joystick en sistemas embebidos permite implementar interfaces de control intuitivas, como la manipulación de coordenadas en una matriz LED o el direccionamiento de un actuador. Su simplicidad, bajo costo y compatibilidad con circuitos de baja tensión lo convierten en un componente común en proyectos de robótica, videojuegos y automatización [10].



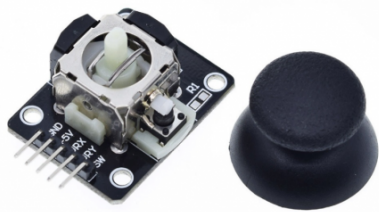


Fig. 9. Joystick analógico de dos ejes con pulsador integrado

### J. Driver L298N

El L298N es un driver de puente H dual que permite controlar la velocidad y el sentido de giro de motores de corriente continua (DC) y motores paso a paso. Este módulo se utiliza en sistemas embebidos para manejar cargas inductivas que requieren más corriente de la que un microcontrolador puede suministrar directamente [11].

Entre sus principales características destacan:

- Capacidad para controlar dos motores DC o un motor paso a paso.
- Corriente máxima de 2 A por canal con protección térmica integrada.
- Control de dirección mediante entradas lógicas IN1–IN4.
- Posibilidad de modulación por ancho de pulso (PWM) para el control de velocidad.
- Operación con voltajes de motor de hasta 46 V.

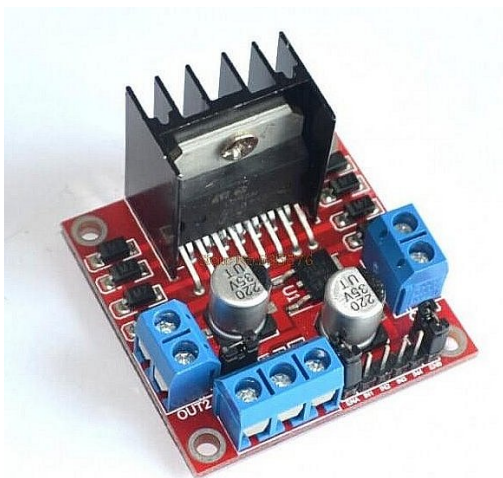


Fig. 10. Driver L298N

## V. PROCEDIMIENTO

A continuación se detallan los pasos seguidos para cumplir con cada apartado del laboratorio.

### A. Parte A

Con respecto al control del plotter de dibujo, este es controlado por dos motores paso a paso y un solenoide, por lo que en primer lugar se definen los pines de control utilizados por el microcontrolador:

- **clk\_x y clk\_y**: generan los pulsos de reloj que hacen avanzar los motores paso a paso.
- **dir\_x y dir\_y**: determinan el sentido de giro de cada motor (derecha/izquierda o arriba/abajo).
- **en\_x y en\_y**: habilitan o deshabilitan los motores.
- **SOL**: controla el solenoide encargado de subir o bajar el lápiz.

Luego se hizo uso de los timers para generar las señales necesarias para los motores. El timer 2 se utiliza para el control del eje X, se configuró en modo PWM rápido, lo que permitió ajustar la velocidad del motor mediante el registro *OCR2A*. Mientras que el timer 2 se utiliza para el eje Y, configurando el modo de comparación con interrupciones. Cada vez que se produce una interrupción, se invierte el estado del *clk\_y*, generando así los pulsos necesarios para el avance de este motor.

Para facilitar el control se realizaron funciones para el desplazamiento en los distintos ejes:

- **mover\_derecha()** y **mover\_izquierda()**: controlan el movimiento del motor del eje X.
- **mover\_arriba()** y **mover\_abajo()**: controlan el movimiento del motor del eje Y.
- **detener()**: desactiva ambos motores.
- **subir\_lapiz()** y **bajar\_lapiz()**: controlan el solenoide para levantar o bajar el lápiz durante el dibujo.

Estas funciones se combinan con retardos temporales *delay\_ms()* para definir la trayectoria de la lapicera.

A partir de las funciones básicas de movimiento, se programan rutinas que dibujan distintas figuras:

Para la cruz y el triángulo se adaptó la lógica utilizada en el plotter anterior para el funcionamiento del nuevo, asegurando así el dibujado de estas figuras.

Con respecto al círculo, se utilizaron bucles for y se variaron los tiempos de delay para crear rectas con distintas pendientes, dando la ilusión de curva. Luego, esto se repitió 4 veces con el fin de cerrar completamente el círculo.

Finalmente, para la rana y la flor, se utilizaron imágenes de referencia, estas pueden verse en las figuras 11 y 12 respectivamente.

### B. Parte B

### C. Parte C

Para el desarrollo de esta parte del laboratorio se desarrolló el control de un motor mediante potenciómetros. Se usan dos potenciómetros uno de referencia que es el que se mueve y activa el motor, cuando el motor hace coincidir el segundo potenciómetro con el de referencia el motor se detiene.

Inicialmente se configuró el microcontrolador (1), activando la comunicación serial mediante UART a 9600 baudios, lo que permitió monitorear en tiempo real los valores de los



Fig. 11. Rana de referencia

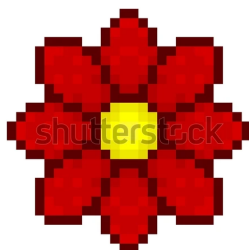


Fig. 12. Flor de referencia

potenciómetros, la dirección del motor y el valor de PWM aplicado. Posteriormente, se configuró el ADC interno para habilitar la lectura continua de dos potenciómetros conectados a los canales analógicos 0 y 1, los cuales representan la referencia deseada y la posición actual del motor, respectivamente.

Seguidamente, se inicializó el Timer1 del microcontrolador en modo PWM rápido con TOP definido por ICR1, habilitando la modulación por ancho de pulso para controlar la velocidad del motor. Los pines de dirección y PWM del motor (PB1, PB2 y PB3) se definieron como salidas digitales, estableciendo la conexión con el driver del motor DC. En cuanto a las conexiones físicas, el motor se conectó a un driver tipo L298N o equivalente, recibiendo señales de dirección desde los pines PB2 y PB3, mientras que la señal PWM proveniente del pin PB1 modulaba la potencia entregada al motor. Los potenciómetros se conectaron a los canales ADC0 y ADC1 para permitir la lectura de la referencia y la posición real del motor, asegurando que las lecturas se mantuvieran dentro del rango de 0 a 1023. Finalmente, la comunicación UART permitió visualizar los datos de manera inmediata en el computador, facilitando la verificación del correcto funcionamiento del sistema.

Durante la ejecución del programa, el microcontrolador realizó lecturas continuas de ambos potenciómetros y calculó la diferencia entre la posición deseada y la actual. Se implementó una zona muerta, de manera que diferencias menores a 10 unidades no activaran el motor, evitando oscilaciones innecesarias. Si la diferencia era positiva, el motor giraba en sentido

horario activando los pines de dirección correspondientes; si era negativa, el motor giraba en sentido antihorario. El valor absoluto de la diferencia se utilizó para calcular el ciclo de trabajo de la señal PWM, regulando así la velocidad del motor de forma proporcional. Se aplicó un límite mínimo de 50 para garantizar un torque suficiente para vencer la inercia y un límite máximo de 255 para evitar saturación y posibles daños al motor.

Simultáneamente, el sistema envió a través de UART los valores de los potenciómetros, el valor de PWM aplicado y la dirección del motor, permitiendo monitorear en tiempo real la operación y realizar ajustes de ser necesario. Durante las pruebas, se verificó que el motor respondiera adecuadamente a los cambios en la posición del potenciómetro de referencia, que la zona muerta funcionara correctamente y que los valores enviados por UART coincidieran con las lecturas reales. El sistema ejecutó de manera continua los ciclos de lectura, cálculo de diferencia, ajuste de dirección y modulación de PWM, manteniendo el motor en la posición deseada y permitiendo una operación estable y controlada.

Para monitorear el desempeño del sistema en tiempo real, se implementó un script en Python utilizando las librerías PySerial y Matplotlib. Este script estableció la conexión serial con el microcontrolador, leyendo los datos enviados por UART y guardándolos simultáneamente en un archivo CSV para su posterior análisis. Las lecturas de los potenciómetros y el valor de PWM se almacenaron en buffers circulares, lo que permitió graficar continuamente los últimos 100 puntos en un gráfico interactivo. Gracias a la actualización dinámica del gráfico, se pudo observar cómo el motor respondía a los cambios de referencia, cómo se aplicaba la zona muerta y cómo variaba la señal PWM en función de la diferencia entre la posición deseada y la actual. Esta visualización facilitó la verificación del control proporcional implementado y permitió ajustar parámetros de forma rápida y precisa.

#### D. Parte D

Para la antepenúltima parte de este laboratorio se implementó un sistema de control de un LED dentro de una matriz RGB de 8x8 utilizando un joystick físico como interfaz de usuario. El objetivo principal consistió en desplazar un LED activo dentro de la matriz según el movimiento del joystick y permitir el cambio aleatorio de color al presionar el botón integrado del mismo.

El sistema se inició configurando el microcontrolador ATmega328P. Se definieron los pines de salida para el control de los LEDs tipo WS2812 y se configuró el pin correspondiente al botón del joystick como entrada con resistencia de pull-up interna. Posteriormente, se inicializó el ADC interno del microcontrolador para permitir la lectura continua de los ejes X y Y del joystick, asignando los canales analógicos 0 y 1 para estas mediciones.

Al encender el sistema, se estableció un LED inicial activo ubicado aproximadamente en el centro de la matriz, con un color predeterminado. Para enviar los valores de color a los LEDs, se implementaron funciones específicas que manejan la

señal serial requerida por los LEDs WS2812. Estas funciones permiten transmitir los bits correspondientes a los valores de rojo, verde y azul de cada LED, asegurando que solo el LED activo mostrara color mientras los demás permanecen apagados. Además, se incluyó una función de reinicio de la señal para garantizar la actualización correcta de toda la matriz.

Durante la ejecución, el microcontrolador realiza lecturas continuas del joystick. Se definió una zona muerta para los ejes X e Y con el fin de filtrar pequeños movimientos o ruidos analógicos que podrían provocar desplazamientos erráticos del LED. Cuando la lectura de un eje excede el umbral de sensibilidad definido, se actualiza la posición del LED en la dirección correspondiente, siempre respetando los límites de la matriz para evitar que el LED se salga del área visible. Una vez detectado un movimiento válido, se actualiza la posición y se muestra inmediatamente en la matriz el LED activo con su color correspondiente.

Adicionalmente, el sistema permite cambiar el color del LED activo de manera aleatoria al presionar el botón del joystick. Para ello, se implementó una función que genera valores aleatorios para los canales de rojo, verde y azul, manteniendo la posición actual del LED dentro de la matriz. Este cambio de color se refleja inmediatamente gracias a la actualización continua de la matriz mediante la función de envío de colores a los LEDs.

El programa principal ejecuta de manera cíclica las funciones de lectura del joystick, detección de presiones del botón y actualización de la matriz. Se incluyeron retardos cortos para asegurar una respuesta fluida y evitar movimientos múltiples no deseados provocados por lecturas rápidas consecutivas del joystick. Además, se utilizó la semilla aleatoria inicial basada en la lectura del ADC para garantizar que los colores generados sean distintos cada vez que se inicia el sistema.

De esta manera, se logró controlar de manera interactiva el movimiento de un LED dentro de la matriz RGB, así como modificar su color de forma aleatoria mediante la interacción física con el joystick, cumpliendo con los objetivos del laboratorio y demostrando la correcta integración entre la lectura analógica, el control digital de los LEDs y la respuesta inmediata del sistema a las acciones del usuario.

#### *E. Parte E*

Para la última parte del laboratorio se implementó una cerradura electrónica inteligente utilizando un microcontrolador ATmega328P y un lector RFID RC522, con el objetivo de controlar el acceso mediante tarjetas o llaveros RFID. El sistema cuenta con una pantalla LCD con interfaz I2C para mostrar mensajes al usuario, dos LEDs indicadores (rojo y verde) para reflejar el estado de acceso y un buzzer que actúa como señal de advertencia sonora. Además, se incorporaron dos botones físicos, uno destinado a actualizar la tarjeta autorizada y otro a borrar la tarjeta registrada.

En primer lugar, se configuraron las conexiones físicas del sistema. El lector RFID se comunicó con el microcontrolador mediante el bus SPI, mientras que la pantalla LCD se conectó

a través del bus I2C. Los LEDs de estado se conectaron a pines digitales del puerto D y los botones de actualización y borrado se configuraron como entradas con resistencias de pull-up internas. El buzzer se conectó a un pin de salida para emitir alertas sonoras cuando fuera necesario.

Posteriormente, se inicializaron los periféricos del microcontrolador, incluyendo SPI para la comunicación con el lector, TWI (I2C) para la pantalla, y UART para enviar información de supervisión a un computador. También se configuró la memoria EEPROM interna para almacenar de manera permanente el UID de la tarjeta autorizada, garantizando que la configuración se mantuviera aún después de apagar el sistema.

Durante la inicialización, la pantalla LCD mostró el mensaje de bienvenida “Bienvenidos al sistema RFID” y se envió el mismo mensaje por UART. Luego, el sistema entró en modo de espera, mostrando “Pasar tarjeta” en la pantalla y notificando por UART que estaba listo para la lectura. Cuando el lector detecta la presencia de una tarjeta, el microcontrolador obtiene su UID y lo compara con el almacenado previamente en la EEPROM. Si el UID coincide, se habilita el acceso: se enciende el LED verde, se muestra “Acceso permitido” en la pantalla y se envía el mismo mensaje por UART. En caso de no coincidir, se enciende el LED rojo y el buzzer emite una alerta breve, mientras el display y la comunicación serial informan “Acceso denegado”.

El sistema permite actualizar la tarjeta autorizada mediante el botón correspondiente. Al presionarlo, la pantalla indica “Acerque nueva tarjeta” y el sistema espera a que se acerque un nuevo llavero RFID. Una vez leído, el UID de la nueva tarjeta se almacena en la EEPROM reemplazando al anterior, se enciende el LED verde y se confirma la operación mediante el mensaje “Nueva tarjeta registrada”, tanto en la pantalla como por UART. Por su parte, el botón de borrado elimina el contenido de la EEPROM asociado a la tarjeta registrada. Al activarlo, la pantalla muestra “Tarjeta borrada” y se envía el mismo mensaje por UART, de manera que mientras no se registre una nueva tarjeta ningún UID será reconocido como válido.

El programa principal se ejecuta en un ciclo continuo que monitorea las tarjetas RFID y el estado de los botones de actualización y borrado. Se implementaron retardos breves entre operaciones para evitar lecturas múltiples no deseadas y garantizar la estabilidad del sistema. De esta manera, se logró integrar de forma efectiva comunicación SPI, I2C, UART y memoria EEPROM, simulando una cerradura electrónica con control de acceso confiable, persistente y completamente interactiva, cumpliendo los objetivos del laboratorio y reforzando los conceptos de manejo de periféricos, memoria no volátil y sistemas embebidos.

## VI. RESULTADOS

#### *A. Parte A*

Durante el desarrollo de esta práctica se logró implementar con éxito el control del plotter de dibujo a través del microcontrolador, utilizando dos motores paso a paso y un solenoide. El sistema respondió adecuadamente a las rutinas programadas,



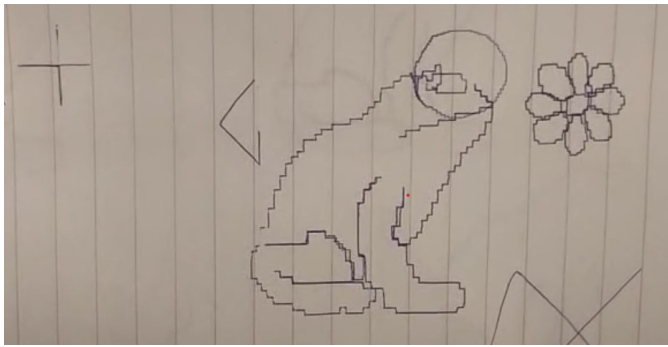


Fig. 13. Figuras dibujadas por el plotter

permitiendo la ejecución de distintos movimientos coordinados en los ejes X e Y, así como el accionamiento preciso del lápiz mediante el solenoide.

En primer lugar, se comprobó el correcto funcionamiento de los pines definidos para el control de los motores y del solenoide. Los pulsos generados por los pines `clk_x` y `clk_y` permitieron el avance de los motores paso a paso, mientras que las señales de dirección posibilitaron el cambio de sentido de giro. Asimismo, las señales de habilitación y el control del pin SOL garantizaron la activación selectiva de los motores y del mecanismo de levantamiento del lápiz.

El uso de los temporizadores fue fundamental para generar las señales de control necesarias. El Timer 2, configurado en modo PWM rápido, permitió ajustar la velocidad del motor correspondiente al eje X mediante el registro OCR2A, mientras que el Timer 1, configurado en modo de comparación con interrupciones, controló el eje Y. En este último caso, se verificó que, con cada interrupción, el pin `clk_y` cambiaba de estado, generando los pulsos necesarios para el desplazamiento del motor.

En cuanto a los resultados gráficos obtenidos, se logró la correcta representación de las figuras programadas. Sin embargo, en la figura 13, por errores de tinta en la lapicera, las figuras tanto del triángulo como de la cruz se encuentran incompletas.

## B. Parte B

## C. Parte C

En esta etapa del laboratorio se evaluó el funcionamiento del control de velocidad de un motor DC mediante modulación por ancho de pulso (PWM). Se registraron las variaciones de tensión y corriente del motor al modificar el ciclo de trabajo, verificando la correspondencia entre el incremento del duty cycle y el aumento de la velocidad angular. Además, se observó la respuesta dinámica del motor ante cambios bruscos de consigna y la estabilidad del sistema en régimen permanente.

Se comenzó con la implementación en físico una vez iniciado el código. Con los pines definidos y los materiales separados se hicieron las conexiones necesarias para el fun-

cionamiento de este, como es posible observar en la siguiente imagen.

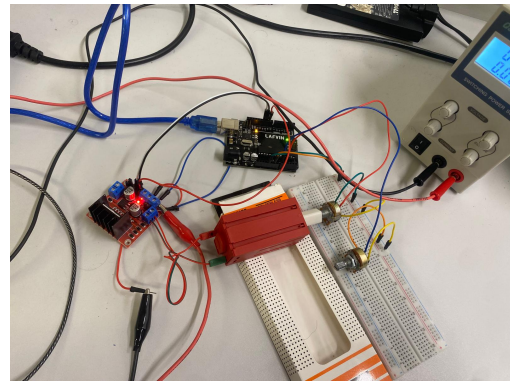


Fig. 14. Implementación motor DC con PWM

Se usaron dos maneras para comprobar el correcto funcionamiento de este apartado, como primera opción se usó la comunicación UART donde mediante puerto serial se mostraba la posición del potenciómetro de referencia, el potenciómetro conectado al motor, el valor del PWM y el sentido de giro, como se muestra en la siguiente imagen.

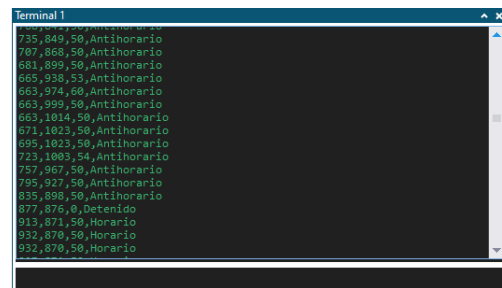


Fig. 15. Funcionamiento con UART

El segundo método de comprobación fue mediante la librería `pyserial`, con esto fue posible obtener gráficas en tiempo real, donde se mostraba el valor del potenciómetro de referencia, el potenciómetro conectado al motor y los pulsos de PWM, la siguiente imagen es el gráfico obtenido con el funcionamiento logrado de este apartado del laboratorio. La tabla [1] ubicada en apartado de anexo (IX) corresponde a los datos tomados para la realización de la gráfica.

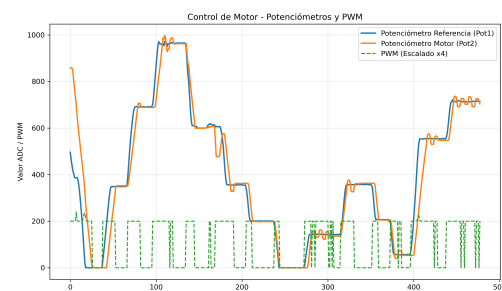


Fig. 16. Gráfica tiempo real motor con PWM

Con la gráfica obtenida se comprobó que a pesar de unas pequeñas oscilaciones el motor fue capaz de igualar el valor del potenciómetro de referencia.

Teniendo la implementación en físico y los resultados obtenidos se pasó a realizar la simulación del problema (17). Se usó el software 'Proteus' y como no fue posible hacer que el motor manejara el potenciómetro estos fueron manejados a mano y así se obtuvo una simulación satisfactoria y cumpliendo con la consigna, en este caso solo se usó la comunicación UART para ver los datos obtenidos (18).

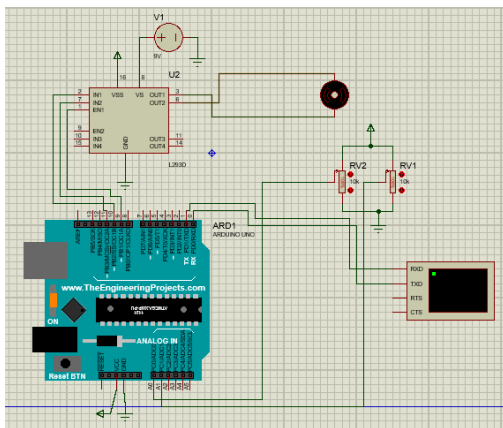


Fig. 17. Simulación manejo motor PWM

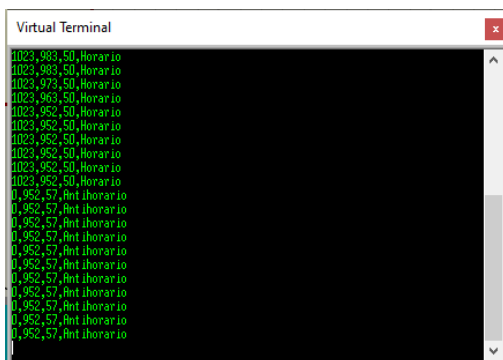


Fig. 18. UART en simulación

#### D. Parte D

En esta etapa se implementó el control de una matriz de LEDs RGB mediante un joystick analógico. A diferencia de las secciones anteriores, esta parte no fue simulada digitalmente, sino implementada directamente en hardware físico. El sistema fue programado para encender un único LED de color variable dentro de la matriz, permitiendo desplazarlo en las cuatro direcciones mediante los ejes analógicos del joystick.

Se tuvo que comprender como estaban conectados los LEDs en la matriz, cual era su patrón para poder realizar el movimiento en los dos ejes de manera correcta, una vez descubierta la conexión en modo lineal se adaptó el código llegando al resultado esperado, el LED iniciaba en las inmediaciones del centro con un color predefinido y era capaz de

moverse entre los ejes, además de un movimiento en diagonal y cambiar de color de manera aleatoria. La siguiente imagen es un fragmento del video de la implementación funcional.

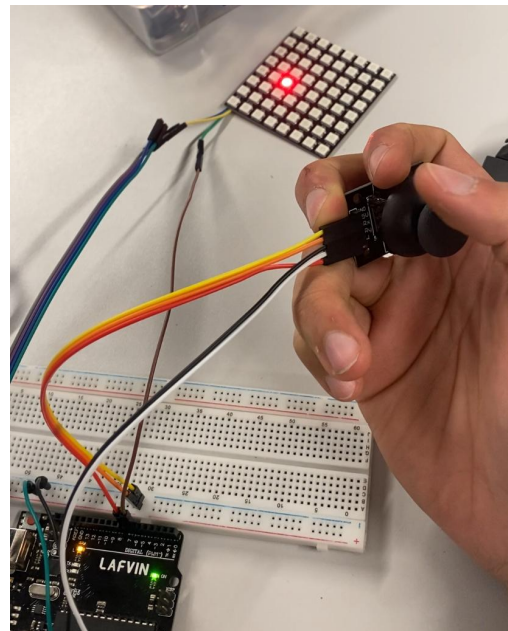


Fig. 19. Matriz con módulo joystick

#### E. Parte E

En la última parte del laboratorio se verificó el correcto funcionamiento del sistema de cerradura electrónica basado en RFID. Se ensayaron múltiples tarjetas para comprobar la detección, validación y respuesta del sistema, tanto en los casos de acceso autorizado como denegado. También se comprobó el almacenamiento y recuperación del UID en memoria EEPROM, la actualización de tarjetas, el borrado de registros y la comunicación de estado mediante la pantalla LCD, LEDs y mensajes enviados por UART.

Como se hizo en el apartado 'C' se comenzó con la implementación en físico. Ya habiendo usado la pantalla LCD con módulo I2C (4) varias veces y entendiendo las librerías brindadas por los profesores el armado del código fue sencillo, usando los mensajes colocados en la consigna. También mediante UART se mostraban en tiempo real lo que ocurría en el sistema. La mayor complicación fue hacer entrar las tres opciones posibles en una misma interfaz, una vez logrado este apartado se completó la consigna de manera muy satisfactoria obteniendo una cerradura electrónica funcional. La siguiente imagen muestra el montaje del sistema RFID.

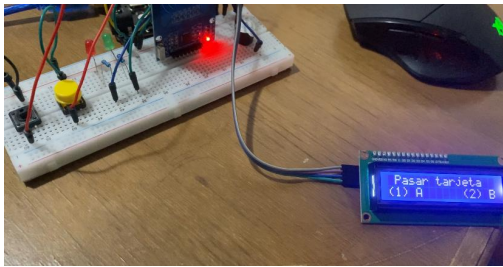


Fig. 20. Montaje sistema RFID

Completada y superada la implementación en físico se realizó la simulación del problema por comodidad se utilizó el software 'PicSimLAB' donde tenía todos los elementos para la realización del apartado. Como no había un sistema de pasado de tarjeta RFID se decidieron usar dos botones con direcciones de memoria guardados, es así que al presionar uno de los botones simulaba que se pasaba un tag o tarjeta con algo cargado. Para esto se hicieron unas modificaciones en el código de librerías haciendo una función que leyera los botones en vez de la tarjeta y es así que se completó la simulación del sistema RFID. La siguiente imagen corresponde a dicha simulación.

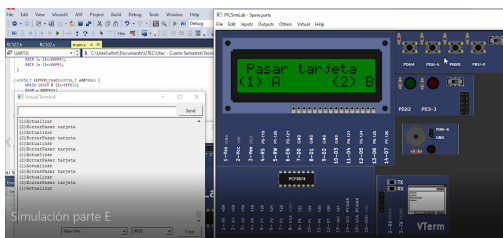


Fig. 21. Simulación RFID

## VII. CONCLUSIÓN

El desarrollo del Laboratorio 3 permitió consolidar de forma práctica los conceptos fundamentales asociados al uso del microcontrolador ATmega328P, la integración de sensores y actuadores, y la comunicación serial entre dispositivos. A través de las distintas partes del trabajo se lograron implementar sistemas embebidos funcionales que abarcan desde el control de movimiento y temperatura hasta la interacción con el usuario y la gestión de acceso mediante RFID.

La implementación del plotter electrónico demostró el correcto manejo de motores paso a paso y la sincronización temporal mediante temporizadores internos. El control del motor con referencia analógica evidenció la importancia del uso del PWM para el desarrollo de sistemas de regulación automática. La matriz RGB controlada por joystick destacó la versatilidad del microcontrolador en tareas de lectura analógica y control digital de periféricos, mientras que la cerradura electrónica RFID integró con éxito múltiples interfaces de comunicación (SPI, I2C y UART), validando el almacenamiento persistente en memoria EEPROM.

En conjunto, las experiencias realizadas fortalecieron la comprensión del diseño modular, la depuración de código y la adaptación de hardware a distintos contextos de control.

Asimismo, se comprobó la relevancia de la documentación estructurada y del trabajo experimental como herramientas para afianzar competencias en electrónica digital, control automático y programación embebida. Finalmente, el laboratorio permitió visualizar la estrecha relación entre teoría y práctica, destacando la capacidad del estudiante para integrar sistemas complejos de forma eficiente, robusta y funcional.

## VIII. BIBLIOGRAFÍA

### REFERENCES

- [1] W. Wolf, *Computers as Components: Principles of Embedded Computing System Design*, 3rd ed., Morgan Kaufmann, 2012.
- [2] Microchip Technology Inc., *ATmega328P: 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*, Datasheet, 2023.
- [3] Microchip Technology Inc., *AVR306: Using the USART in AVR Microcontrollers*, Application Note, 2021.
- [4] J. García and M. Pérez, *Automatización y Control Industrial*, 2nd ed., Madrid: Alfaomega, 2020.
- [5] D. Flores, *Interfacing LCDs via I2C Bus: Using the PCF8574 with HD44780 Displays*, Texas Instruments Application Note, 2020.
- [6] Microchip Technology Inc., *AVR101: High Endurance EEPROM Storage in AVR Microcontrollers*, Application Note, 2021.
- [7] Worldsemi, “WS2812 Intelligent Control LED Datasheet,” 2022. [Online]. Available: <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>
- [8] Texas Instruments, “LM35 Precision Centigrade Temperature Sensors,” Datasheet, 2020. [Online]. Available: <https://www.ti.com/lit/ds/symlink/lm35.pdf>
- [9] Microchip Technology Inc., “AVR130: Setup and Use of AVR Timers,” Application Note, 2021. [Online]. Available: <https://ww1.microchip.com>
- [10] SparkFun Electronics, “Analog 2-Axis Thumb Joystick Hookup Guide,” 2022. [Online]. Available: <https://learn.sparkfun.com/tutorials/analog-joystick-hookup-guide>
- [11] STMicroelectronics, *L298N: Dual H-Bridge Motor Driver Datasheet*, 2021. [Online]. Disponible en: <https://www.st.com/en/motor-drivers/l298.html>
- [12] Adafruit, “Heating Element Control with PWM and Temperature Feedback,” 2022. [Online]. Disponible en: <https://learn.adafruit.com/thermistor/using-with-heaters>

## IX. ANEXO

- [1] Link archivo con los datos obtenidos para el apartado C