

Microcontroladores: Laboratorio 1

1st Mateo Lecuna

Ingeniería en Mecatrónica

Universidad Tecnológica (UTEC)

Fray Bentos, Uruguay

mateo.lecuna@estudiantes.utec.edu.uy

2nd Mateo Sanchez

Ingeniería en Mecatrónica

Universidad Tecnológica (UTEC)

Maldonado, Uruguay

mateo.sanchez@estudiantes.utec.edu.uy

Resumen—Se presenta el desarrollo de cinco sistemas embebidos implementados sobre el microcontrolador ATmega328P, correspondientes al Laboratorio 3 de la unidad curricular *Tecnologías de Microprocesamiento*. Los ejercicios abordaron distintos escenarios de control digital y adquisición de señales, incluyendo un plóter cartesiano de dos ejes, un sistema de control de temperatura, un control de motor por referencia analógica, una matriz RGB controlada mediante joystick, y una cerradura electrónica basada en lector *RFID RC522*.

Cada proyecto integró periféricos de entrada/salida, sensores analógicos, comunicación serial UART e interfaces de visualización (LCD I2C y Python/MATLAB para gráficas). Se implementaron estrategias de control mediante modulación por ancho de pulso (PWM) y lógica secuencial programada en lenguaje C. Los resultados obtenidos demostraron el correcto funcionamiento de los sistemas tanto en simulación como en hardware físico, validando el aprendizaje y aplicación práctica de los conceptos de microprocesamiento y control embebido.

Keywords: ATmega328P, sistemas embebidos, modulación PWM, control de motor, joystick, matriz RGB, RFID, UART, microcontroladores.

I. INTRODUCCIÓN

El presente laboratorio tuvo como objetivo aplicar los conocimientos adquiridos en la unidad curricular *Tecnologías de Microprocesamiento*, a través del diseño, programación e implementación de distintos sistemas embebidos basados en el microcontrolador **ATmega328P**. A partir de este dispositivo se desarrollaron cinco ejercicios experimentales que integran control digital, manejo de periféricos, adquisición de señales analógicas, y comunicación serial.

Los ejercicios planteados corresponden a los siguientes problemas:

- **Problema A – Control de Plotter:** Implementación de un sistema de control para un plóter cartesiano de dos ejes, empleando motores paso a paso y una válvula solenoide para el trazado de figuras. El objetivo principal consiste en coordinar los movimientos de los ejes X e Y respetando los límites definidos por sensores, logrando la representación precisa de figuras geométricas.
- **Problema B – Sistema de Control de Temperatura:** Desarrollo de un sistema de regulación térmica mediante el sensor LM35. El microcontrolador mide

la temperatura en intervalos regulares y controla un calefactor y un ventilador por modulación PWM, con posibilidad de ajustar el punto medio de temperatura a través de un menú UART.

- **Problema C – Control de Motor:** Implementación de un sistema de control de posición analógico en el que dos potenciómetros determinan la referencia y la posición actual de un motor de corriente continua. El microcontrolador regula el giro y la velocidad del motor mediante una señal PWM, buscando igualar ambos valores. El comportamiento se monitorea por puerto serial y se analiza mediante gráficas de evolución temporal.
- **Problema D – Matriz RGB con Joystick:** Diseño de una interfaz interactiva en la que el movimiento del joystick desplaza un LED encendido dentro de una matriz RGB. El sistema detecta las direcciones de movimiento mediante entradas analógicas (ADC) y genera un cambio de color aleatorio al presionar el pulsador integrado en el joystick.
- **Problema E – Cerradura RFID:** Construcción de una cerradura electrónica inteligente basada en un lector *RC522 RFID*, con almacenamiento del identificador autorizado en la memoria EEPROM, interfaz de visualización LCD I2C y señalización por LEDs y buzzer. El sistema gestiona el registro, validación y actualización de tarjetas, así como la comunicación UART para monitoreo externo.

Cada uno de estos ejercicios aborda distintos aspectos del control digital, incluyendo la comunicación UART, la modulación por ancho de pulso (PWM), la lectura de señales analógicas mediante el ADC interno y el manejo de periféricos de entrada/salida (I/O). En conjunto, el laboratorio permitió consolidar la comprensión del funcionamiento del microcontrolador ATmega328P y su aplicación en proyectos de automatización y control embebido.

En las siguientes secciones se presenta el desarrollo teórico, metodológico y experimental de los sistemas implementados, junto con los resultados obtenidos y las conclusiones correspondientes.

II. MARCO TEÓRICO

II-A. Plotter

II-B. Sistema de Control de Temperatura

II-C. Control de Motor

El sistema de control de motor implementado tiene como propósito igualar el valor leído por un potenciómetro de referencia con el valor obtenido de otro potenciómetro acoplado mecánicamente al eje de un motor de corriente continua (DC). Para lograrlo, el microcontrolador **ATmega328P** compara ambas señales analógicas, determina la dirección de giro correspondiente y ajusta la velocidad del motor mediante modulación por ancho de pulso (*PWM*), utilizando un puente H **L298N** como etapa de potencia.

Motor de corriente continua (DC): El motor DC convierte energía eléctrica en mecánica mediante la interacción entre el campo magnético del estator y la corriente en el rotor. La velocidad de giro es proporcional al voltaje aplicado, mientras que el sentido de rotación depende de la polaridad. En este sistema, el motor opera dentro de un rango de alimentación de 6–12 V, con un comportamiento estable a partir de un ciclo útil mínimo equivalente a una señal PWM de aproximadamente 180/255. Este valor fue determinado experimentalmente, ya que con ciclos menores (120/255) el par resultaba insuficiente para vencer la carga impuesta por el acoplamiento con el potenciómetro del eje.

Puente H L298N: El circuito integrado L298N permite invertir la polaridad aplicada al motor mediante el control de cuatro transistores dispuestos en configuración de puente H. El microcontrolador gobierna las entradas IN1 e IN2 para establecer el sentido de giro, y la entrada ENA se conecta a un pin PWM para modular la velocidad. Esta arquitectura posibilita un control bidireccional eficiente, evitando el uso de relevadores o conmutadores mecánicos. El módulo integra además diodos de rueda libre para protección ante corrientes inversas generadas por la inductancia del motor.

Potenciómetros de referencia y realimentación: Se emplearon dos potenciómetros lineales de 10 k Ω . El primero, conectado al pin analógico A0, define la posición o velocidad de referencia (*setpoint*), mientras que el segundo, vinculado físicamente al eje del motor y conectado al pin A1, mide la posición actual. Ambos valores son convertidos por el conversor analógico–digital (ADC) de 10 bits del ATmega328P, proporcionando una resolución suficiente para el control analógico proporcional. Para mejorar la estabilidad de lectura, cada medición promedia cuatro muestras consecutivas, lo que reduce el efecto del ruido eléctrico.

Control por modulación PWM: El ATmega328P genera la señal PWM en el pin OC1A (D9) mediante el temporizador interno Timer1 configurado en modo Fast PWM de 8 bits, con prescaler $N = 1$, alcanzando una frecuencia de conmutación de aproximadamente 62.5 kHz. Este valor resulta inaudible para el oído humano y proporciona una respuesta suave en el motor. En el código se implementó un control proporcional simple que ajusta el ciclo útil del PWM en función de la diferencia entre los valores de referencia y

realimentación. Cuando la diferencia es menor a una zona muerta de 10 unidades ADC, el sistema detiene el motor y apaga los indicadores de dirección.

Comunicación serial y visualización: Durante la ejecución, el sistema transmite por UART los valores de ambos potenciómetros, el sentido de giro y el valor del PWM aplicado, en formato CSV (*ref, act, pwm, dir*). Esto permite representar las variables en tiempo real mediante Python o MATLAB, facilitando el análisis de la evolución temporal y la respuesta del control.

Observación experimental del umbral de arranque:

Durante las pruebas se determinó empíricamente un umbral mínimo de ciclo útil para garantizar el arranque bajo carga. Sin acoplamiento mecánico, el motor iniciaba la rotación con $PWM \approx 120/255$; sin embargo, al acoplar el potenciómetro al eje, el incremento de par resistente elevó el umbral. Tras incrementos sucesivos, se fijó un mínimo operativo de **180/255**, que aseguró arranque fiable y evitó bloqueos intermitentes en transitorios de baja referencia. Este valor se adoptó como límite inferior del control para preservar la robustez del sistema.

Detalles de implementación y consideraciones prácticas: El PWM se generó en el pin D9 (OC1A) mediante Timer1 en modo Fast PWM de 8 bits, con prescaler $N = 1$ (frecuencia $\approx 62,5$ kHz), evitando ruido audible en el motor. Dado que el L298N es un puente H con transistores Darling-ton, frecuencias tan elevadas pueden incrementar pérdidas de conmutación; en caso de calentamiento, es razonable reducir la frecuencia (por ejemplo, prescaler $N = 8$, $\approx 7,8$ kHz) sin afectar perceptiblemente el control. Para la adquisición analógica se utilizó el ADC interno a 10 bits con prescaler 64 (≈ 250 kHz) y promediado de cuatro muestras por canal, logrando lecturas estables a 1 kHz de lazo. Si se prioriza la precisión de 10 bits, el prescaler 128 (≈ 125 kHz) es una alternativa acorde a la recomendación del fabricante.

En síntesis, este módulo combina los principios de control analógico con el manejo digital de periféricos del microcontrolador, integrando adquisición de señales (ADC), procesamiento de control proporcional y generación de salida PWM, bajo una estructura de hardware sencilla y de bajo costo.

II-D. Matriz RGB con Joystick

Este módulo implementa una interfaz interactiva en la que un *cursor* luminoso (un único LED encendido) se desplaza sobre una matriz 8 \times 8 de LEDs RGB direccionables **WS2812B**. La posición se controla con un *joystick analógico* (dos potenciómetros ortogonales) y un pulsador que, al activarse, asigna un color RGB pseudoaleatorio al LED activo. El sistema integra: (i) adquisición analógica por ADC, (ii) lectura digital del pulsador con resistencia de *pull-up* interna, y (iii) generación del protocolo de temporización de los WS2812B directamente por software (*bit-banging*), respetando el orden de bytes **GRB** y la ventana de *reset* ($\sim 80 \mu s$) para el *latch* del tren de datos. :contentReferen-
ce[oaicite:0]index=0

Joystick analógico y umbrales de decisión: Cada eje del joystick se lee mediante el conversor A/D de 10 bits del

ATmega328P, empleando V_{cc} como referencia y prescaler 64 (frecuencia de reloj del ADC ≈ 250 kHz), lo que otorga una resolución de 1024 niveles por canal. Se define un punto medio en $MID = 512$ y una *zona muerta* de amplitud $DZ = 90$ cuentas, generando umbrales $TH_{LO} = 512 - 90$ y $TH_{HI} = 512 + 90$. Valores por debajo/encima de esos umbrales se interpretan como movimientos discretos en $\{\uparrow, \downarrow, \leftarrow, \rightarrow\}$ (una celda por evento). Para evitar repeticiones espurias por ruido y velocidad de muestreo, se usa un *cooldown* entre pasos. En la implementación, el eje **X** está invertido (*mapping* físico), de modo que desplazamientos a un lado producen el índice decreciente y al contrario para el otro. :contentReference[oaicite:1]index=1

Pulsador con entrada digital y cambio de color: El pulsador del joystick (SW) se conecta a PD2 como entrada con *pull-up* interno (*activo en bajo*). Al detectar un flanco de pulsación, el sistema genera un nuevo color mediante un generador pseudoaleatorio *LFSR* de 8 bits, actualizando los tres canales (R, G, B) del LED activo y registrando por UART el nuevo triplete RGB.

Matriz WS2812B y protocolo GRB a 800 kHz: Los WS2812B integran controlador por LED, por lo que *no requieren multiplexado externo*. La cadena de 64 LEDs se modela como un arreglo lineal de longitud $N = 64$ (8×8), direccionado por el índice $i \in [0, 63]$. Para transmitir color, se envían 24 bits por LED en orden **G-R-B**, con tiempos de T_{0H} , T_{0L} , T_{1H} , T_{1L} compatibles con 800 kHz. Durante el envío se **deshabilitan interrupciones** (`cli()/sei()`) para garantizar la precisión de ciclo, y se deja un $\sim 80 \mu s$ de *reset* al final para *latch*. Además, se aplica una atenuación global ($BR = 80/255$) mediante una escala lineal por canal antes de transmitir el frame completo. :contentReference[oaicite:3]index=3

Comunicación UART para telemetría: El sistema inicializa la UART (9600 bps, 8N1) y reporta eventos (p.ej., cambio de color) con mensajes formateados, lo que facilita la verificación y documentación experimental desde consola y/o captura en PC.

Lógica de movimiento en retícula 8×8 : El índice i se actualiza con reglas de borde para impedir *wrap-around* horizontal (columnas 0 y 7) y vertical (filas 0 y 7). Se emplea un *cooldown* de pasos para dar una cadencia estable al desplazamiento con el *stick* sostenido y un retardo de refresco de ~ 10 ms por iteración, asegurando una respuesta perceptualmente fluida del cursor. Una vez actualizado i , se limpia el *framebuffer* y se enciende únicamente el LED de índice i con el color actual antes de transmitir el arreglo completo por PB0. :contentReference[oaicite:5]index=5

En síntesis, este ejercicio demuestra la integración entre adquisición analógica (ADC con umbrales y zona muerta), manejo de entradas digitales (botón con *pull-up* y detección de flanco), generación de color pseudoaleatorio, y transmisión temporalmente crítica del protocolo WS2812B por *bit-banging*, todo sobre el **ATmega328P**.

II-E. Cerradura RFID

El sistema de cerradura electrónica implementado permite la identificación y validación de tarjetas mediante la tecnología **RFID** (*Radio Frequency Identification*), utilizando el módulo **MFRC522** como lector principal. El objetivo del experimento es diseñar un sistema de acceso seguro que reconozca credenciales válidas, gestione nuevas altas o bajas y brinde retroalimentación visual mediante una pantalla LCD I2C, LEDs indicadores y comunicación UART.

Módulo RC522 y principio de funcionamiento: El módulo MFRC522 opera en la banda de 13.56 MHz y permite la lectura y escritura de etiquetas o tarjetas compatibles con el estándar **ISO/IEC 14443A** (MIFARE). Su funcionamiento se basa en la inducción electromagnética: el lector genera un campo alterno que alimenta la antena de la tarjeta pasiva, permitiendo el intercambio de información. El microcontrolador se comunica con el RC522 a través del bus **SPI** (pines MOSI, MISO, SCK y SS), enviando comandos y recibiendo respuestas estructuradas en registros internos del chip.

El código implementa una rutina de inicialización (`RC522_Init()`) que configura los registros de temporización y tasa de bits, y una función de detección (`RC522_Request()`) seguida de la lectura del UID con `RC522_AntiColl()`. El UID capturado se compara con los almacenados en memoria EEPROM para determinar si la tarjeta es válida. En caso positivo, se activa la secuencia de apertura, y en caso negativo, el sistema emite una señal de error visual y sonora.

Comunicación SPI: El bus **SPI** (Serial Peripheral Interface) fue configurado en modo maestro ($MSTR=1$) y frecuencia de reloj $F_{osc}/8$, lo que proporciona una velocidad de transferencia de 2 MHz para una operación confiable con el módulo RC522. La comunicación es síncrona y de tipo *full duplex*, utilizando los pines PB3 (MOSI), PB4 (MISO), PB5 (SCK) y PB2 (SS) del ATmega328P.

Almacenamiento en EEPROM: Para conservar las tarjetas autorizadas incluso tras el apagado, se emplea la memoria EEPROM interna del microcontrolador. Cada UID leído se guarda en posiciones consecutivas, permitiendo almacenar múltiples usuarios. El código utiliza las funciones de lectura y escritura de EEPROM estándar de AVR (`eeeprom_read_byte`, `eeeprom_write_byte`) y gestiona el espacio mediante un índice de direcciones. Se implementan comandos UART para listar, eliminar o limpiar las tarjetas registradas, lo que facilita la administración del sistema sin reprogramar el microcontrolador.

Interfaz LCD I2C: El sistema incorpora una pantalla LCD de 16×2 caracteres con interfaz **I2C** basada en el expansor **PCF8574**, la cual permite reducir el número de pines utilizados. Esta interfaz se encarga de mostrar los estados del sistema: detección de tarjeta, modo de programación, mensajes de autorización o error, y menús de administración. Los datos se envían en formato de 4 bits con temporización controlada, y la dirección esclava del módulo corresponde a $0x27$.

Retroalimentación visual y sonora: El estado de la cerradura se indica mediante dos LEDs y un buzzer piezo-eléctrico. El LED verde confirma una tarjeta válida y activa brevemente el buzzer, mientras que el LED rojo se enciende ante un intento no autorizado. Esta respuesta multimodal brinda una interfaz intuitiva al usuario y permite verificar el funcionamiento sin necesidad de monitorear el puerto serie.

Comunicación UART y monitoreo externo: El microcontrolador establece una comunicación serial a 9600 bps (8N1) para registro de eventos, depuración y control remoto del sistema. Mediante esta interfaz se envían mensajes informativos como la detección de tarjetas, UID leído, estado del sistema y comandos de usuario. Las funciones `uart_send()`, `uart_receive()` y `uart_print()` facilitan la interacción con una terminal o aplicación de monitoreo externa, lo que resulta útil durante la programación y verificación del funcionamiento.

Arquitectura general del sistema: El flujo principal se estructura en un *autómata de estados finitos (FSM)*, donde cada estado corresponde a una etapa del proceso: INICIO, ESPERA, PROGRAMACIÓN y VERIFICACIÓN. Los eventos de entrada (detección de tarjeta, pulsación de botones o comandos UART) disparan las transiciones correspondientes. Este enfoque modular permite una gestión ordenada del control lógico y facilita la expansión futura del sistema, por ejemplo, agregando autenticación por contraseña o múltiples niveles de acceso.

En conjunto, el sistema de cerradura RFID combina varios subsistemas de hardware y software: la identificación inalámbrica mediante el RC522, la comunicación serial SPI y UART, la visualización por I2C, y el almacenamiento persistente en EEPROM, integrados todos bajo el control del microcontrolador ATmega328P.

III. METODOLOGÍA

- III-A. *Plotter*
- III-B. *Sistema de Control de Temperatura*
- III-C. *Control de Motor*
- III-D. *Matriz RGB con Joystick*
- III-E. *Cerradura RFID*

IV. RESULTADOS

- IV-A. *Plotter*
- IV-B. *Sistema de Control de Temperatura*
- IV-C. *Control de Motor*
- IV-D. *Matriz RGB con Joystick*
- IV-E. *Cerradura RFID*

V. CONCLUSIONES

- V-A. *Plotter*
- V-B. *Sistema de Control de Temperatura*
- V-C. *Control de Motor*
- V-D. *Matriz RGB con Joystick*
- V-E. *Cerradura RFID*

REFERENCIAS

- [1] STMicroelectronics, *L298 — Dual Full-Bridge Driver (H-Bridge) — Datasheet*, 2022, puente H utilizado para el control bidireccional del motor DC. [Online]. Available: <https://www.st.com/resource/en/datasheet/l298.pdf>
- [2] Electronics Tutorials. H-bridge motor control theory. Conceptos de puente H y conmutación de polaridad. [Online]. Available: <https://www.electronics-tutorials.ws/blog/h-bridge.html>
- [3] SparkFun Electronics. Pulse width modulation (pwm) tutorial. Fundamentos de modulación por ancho de pulso en microcontroladores. [Online]. Available: <https://learn.sparkfun.com/tutorials/pulse-width-modulation>
- [4] Adafruit Industries. (2014) Brushed dc motors — adafruit motor selection guide. Conceptos de motores DC, control y selección de componentes. [Online]. Available: <https://learn.adafruit.com/adafruit-motor-selection-guide/dc-motors>
- [5] NXP Semiconductors, *PCF8574/PCF8574A — Remote 8-bit I²C I/O Expander*, 2023, datasheet oficial del expansor I/O para interfaz I2C de LCD. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf
- [6] SparkFun Electronics. (2017) Thumb joystick hookup guide. Funcionamiento del joystick analógico, ejes, SW y lectura por ADC. [Online]. Available: <https://learn.sparkfun.com/tutorials/thumb-joystick-hookup-guide>
- [7] ——. (2016) Rfid basics. Introducción a RFID, fundamentos y flujo de lectura UID. [Online]. Available: <https://learn.sparkfun.com/tutorials/rfid-basics/all>

VI. ANEXOS