

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 diseño e implementación de cuatro subsistemas sobre el microcontrolador ATmega328P: (1) un *plotter* cartesiano programado en C, (2) un selector/detector de color que explora un círculo cromático, (3) un piano electrónico, y (4) una cerradura electrónica. Cada módulo ejerce competencias clave de sistemas embebidos: planificación de trayectorias y temporización (*plotter*), adquisición/filtrado analógico y mapeo a acciones (LDR→servo→LED), generación de señales y lectura confiable de entradas (piano), y control secuencial por máquina de estados con validación de credenciales (cerradura). Se verificó el funcionamiento estable de los módulos, con lecturas analógicas consistentes, posicionamiento del servo acorde a la detección, generación de tonos precisa y lógica de acceso confiable. Como líneas de mejora se identifican: perfiles de movimiento (trapezoidal/S-curve) para el *plotter*, calibración y normalización de la LDR frente a iluminación ambiente, antirrebote/antighosting en el teclado del piano, y mecanismos adicionales de seguridad en la cerradura (bloqueo por intentos, almacenamiento en EEPROM).

Keywords: ATmega328P, sistemas embebidos, lenguaje C, *plotter* cartesiano, detección de color, servomotor, piano electrónico, cerradura electrónica.

I. INTRODUCCIÓN

Objetivos específicos

II. MARCO TEÓRICO

II-A. Piano

El piano electrónico se basa en el microcontrolador ATmega328P, encargado de leer las entradas digitales provenientes de pulsadores y generar las notas musicales correspondientes a través de un buzzer piezoeléctrico. Para la síntesis de sonido, se hace uso de señales de modulación por ancho de pulso (PWM), configuradas mediante los temporizadores internos del microcontrolador, permitiendo así obtener frecuencias precisas asociadas a cada nota musical.

El sistema implementa un conjunto de 12 pulsadores, cada uno asignado a una nota de la escala cromática (Do, Do#, Re, Re#, Mi, Fa, Fa#, Sol, Sol#, La, La#, Si). Además, se integran dos pulsadores adicionales para modificar la octava activa, lo que amplía la capacidad tonal del instrumento sin aumentar significativamente el número de entradas físicas.

El buzzer piezoeléctrico utilizado actúa como transductor electroacústico, recibiendo la señal PWM generada por el

ATmega328P y transformándola en vibraciones audibles. El uso de resistencias pull-up internas en los pines de entrada digital simplifica el cableado, evitando la necesidad de resistencias externas para los pulsadores.

Por otra parte, la inclusión de la comunicación serial mediante UART (Universal Asynchronous Receiver-Transmitter) permite la selección de canciones predefinidas almacenadas en memoria. De esta forma, el sistema no solo funciona como piano manual, sino también como reproductor automático de melodías programadas.

II-B. Cerradura electrónica

III. METODOLOGÍA

III-A. Materiales a utilizar

- Microcontrolador ATmega328P (plataforma Arduino UNO).
- 12 pulsadores para las notas de la escala cromática.
- 2 pulsadores adicionales para el control de octavas.
- Buzzer piezoeléctrico.
- Resistencias pull-up internas configuradas por software.
- Conexiones con jumpers y protoboard.

III-B. Plotter

III-C. Seleccionador de colores

III-D. Piano

III-D1. Diseño del sistema: El sistema se estructuró en torno a los siguientes bloques principales:

1. **Lectura de pulsadores:** Se configuraron los pines digitales como entradas con resistencias pull-up internas. El estado lógico bajo indica que la tecla fue presionada.
2. **Generación de notas:** Utilizando los temporizadores del ATmega328P en modo PWM, se programaron las frecuencias correspondientes a cada nota musical. Las notas se definieron en una tabla para facilitar su acceso en el programa.
3. **Cambio de octava:** Se reservaron dos pulsadores adicionales para aumentar o disminuir la octava activa. Esto permite variar la frecuencia base de todas las notas según la octava seleccionada.
4. **Reproducción de canciones:** La comunicación UART se estableció para recibir comandos externos y activar la ejecución de melodías predefinidas. Estos comandos permiten conmutar entre modo piano manual y modo automático.

III-D2. Obtención y transcripción de melodías: Para la programación de las canciones predefinidas, fue necesario obtener previamente las secuencias de notas, duraciones y octavas de cada una.

La primera melodía, denominada “*Asesina*”, fue extraída a partir del video de YouTube [1]. Para identificar las notas y sus duraciones se utilizó una aplicación de piano roll disponible en App Store, con la cual se reprodujo la melodía y se registraron manualmente las notas en una hoja de referencia, indicando la octava correspondiente y el tiempo de ejecución de cada una. Posteriormente, estos valores fueron transferidos al código fuente en forma de arreglos.

La segunda melodía, correspondiente al tema principal de *Super Mario Bros*, se obtuvo a partir de la partitura disponible en el portal MuseScore [2]. Dicha fuente incluía además una simulación visual en piano roll, lo que facilitó la identificación de las notas y su duración sin necesidad de realizar el reconocimiento auditivo de las notas, como se hizo con la primera canción.

En el documento de evidencias anexo se incluyen las hojas manuscritas utilizadas en la transcripción de ambas canciones, junto con una tabla que relaciona las notas musicales con su índice numérico dentro del arreglo de notas

del programa. No fue posible incluir el archivo PDF con la partitura de la canción de *Super Mario Bros* debido a que requiere una suscripción paga para su descarga, pero, sin embargo, accediendo al enlace indicado en la bibliografía es posible visualizar la partitura y la simulación en piano roll mencionada anteriormente.

III-E. Cerradura electrónica

IV. RESULTADOS

IV-A. *Plotter*

IV-B. *Seleccionador de colores*

IV-C. *Piano*

El sistema desarrollado logró cumplir satisfactoriamente los objetivos propuestos. Se implementó un piano electrónico basado en el microcontrolador ATmega328P, capaz de reproducir las 12 notas de la escala cromática mediante pulsadores individuales. Cada tecla activa la generación de una frecuencia específica a través del buzzer pasivo, logrando un sonido claro y fácilmente distinguible entre notas.

La comunicación UART fue implementada correctamente, permitiendo el control remoto del sistema desde una interfaz serial. Se configuraron tres comandos principales: C1 y C2, utilizados para la reproducción de dos canciones predefinidas almacenadas en memoria, y el comando PIANO, que devuelve al modo manual de ejecución por teclas. El comando STOP no fue implementado debido a limitaciones de tiempo, aunque el sistema mantiene estabilidad y correcta respuesta durante la ejecución de las melodías.

Durante las pruebas se observó un comportamiento estable en la reproducción de las notas y una respuesta inmediata ante la pulsación de teclas. El buzzer pasivo entregó una calidad sonora adecuada, permitiendo distinguir correctamente las canciones y las frecuencias individuales. No se presentaron problemas de latencia ni de resonancia indebida, confirmando la correcta configuración del temporizador para la generación de las señales PWM.

De forma general, el desempeño del sistema fue considerado satisfactorio, tanto en el modo de ejecución manual como en el automático. Las canciones predefinidas se reprodujeron de forma fluida y reconocible, evidenciando un correcto manejo de tiempos y frecuencias en la modulación del sonido.

IV-D. *Cerradura electrónica*

V. CONCLUSIONES

V-A. *Plotter*

V-B. *Seleccionador de colores*

V-C. *Piano*

El desarrollo del piano electrónico permitió aplicar conceptos fundamentales del manejo de periféricos del microcontrolador ATmega328P, en particular el uso de temporizadores para la generación de señales PWM y la comunicación UART para el intercambio de comandos externos.

El sistema demostró un funcionamiento estable y confiable, logrando la reproducción precisa de las 12 notas musicales y una correcta interpretación de las canciones predefinidas. La calidad del sonido obtenida mediante el buzzer fue adecuada, con una frecuencia de salida clara y sin distorsiones perceptibles.

La implementación de la comunicación serial aportó una capa adicional de control, permitiendo al usuario seleccionar canciones o volver al modo manual de manera sencilla.

Si bien no se desarrolló el comando STOP ni el cambio de octavas, los objetivos principales del ejercicio fueron alcanzados con éxito.

En términos generales, este ejercicio evidenció una correcta integración entre hardware y software, así como una adecuada gestión de temporización y respuesta a eventos. Como líneas de mejor, se puede proponer incorporar un sistema de control de volumen o modulación, implementar el comando de detención, y explorar el uso de interrupciones para optimizar la detección de teclas y la eficiencia del sistema.

V-D. *Cerradura electrónica*

REFERENCIAS

- [1] CASTIGADOSUY. (2023) Canción asesina [video]. Accedido el 18 de octubre de 2025. Fuente utilizada para la transcripción de la primera melodía. [Online]. Available: https://www.youtube.com/watch?v=ZSH1W_cJ31Q
- [2] WildMojarras. (2020) Super mario bros theme [score]. Accedido el 18 de octubre de 2025. Fuente utilizada para la transcripción de la segunda melodía. [Online]. Available: <https://musescore.com/user/27687306/scores/4913846>
- [3] Microchip Technology Inc. Atmega328p datasheet. Documento técnico del microcontrolador utilizado en todos los ejercicios del laboratorio. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- [4] circuito.io. (2018) Arduino uno pinout diagram. Referencia de pines y funciones del microcontrolador ATmega328P en la placa Arduino Uno. [Online]. Available: <https://www.circuito.io/blog/arduino-uno-pinout/>
- [5] Microchip Community (AVR Freaks). Avr freaks — comunidad de desarrolladores avr. Foros técnicos y soluciones prácticas sobre AVR. [Online]. Available: <https://www.avrfreaks.net/>
- [6] Carpeta del laboratorio (google drive). Carpeta compartida del laboratorio con evidencias y videos de funcionamiento. [Online]. Available: <https://drive.google.com/drive/u/0/folders/1fP0aILozXeapRgDPDNWT1TRhAYr1PPPT>
- [7] Laboratorio de Microcontroladores, UTEC, "Repositorio de laboratorio de microcontroladores (tec.micro)," <https://github.com/MateoLecuna/Tec.Micro>, 2025, repositorio de código y versiones de los ejercicios del laboratorio.
- [8] Mateo Lecuna. (2025) Evidencias del laboratorio 2 — repositorio tec.micro. Carpeta con videos, imágenes y registros del funcionamiento de los sistemas del Laboratorio 2. Accedido el 18 de octubre de 2025. [Online]. Available: <https://github.com/MateoLecuna/Tec.Micro/tree/main/Laboratorio%202/Evidencias>

VI. ANEXOS

VI-A. Esquemas de conexión

VI-A1. Piano electrónico: En la Figura 1 se muestra el esquema de conexión del piano electrónico, elaborado en Tinkercad®. Se pueden observar las conexiones entre el microcontrolador ATmega328P, los pulsadores, y el buzzer piezoeléctrico pasivo EMX-7T05SP.

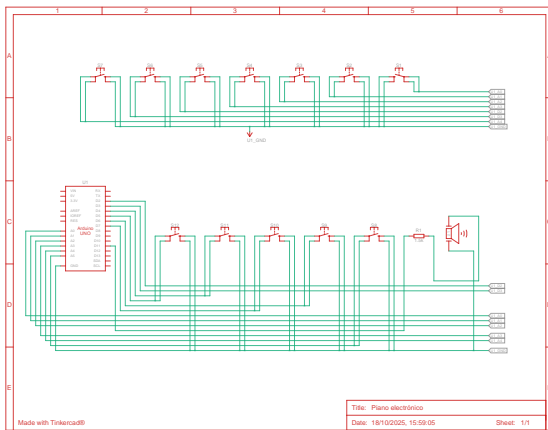


Figura 1. Esquema de conexión del piano electrónico. Elaboración propia en Tinkercad®.

VI-A2. Cerradura electrónica: En la Figura 2 se presenta el esquema de conexión del sistema de cerradura electrónica. El circuito fue diseñado en Tinkercad® y muestra la interconexión entre el microcontrolador ATmega328P, el teclado matricial 4×4, la pantalla LCD 16×2 con interfaz I²C, los LEDs indicadores (rojo y verde) y el buzzer de señalización.

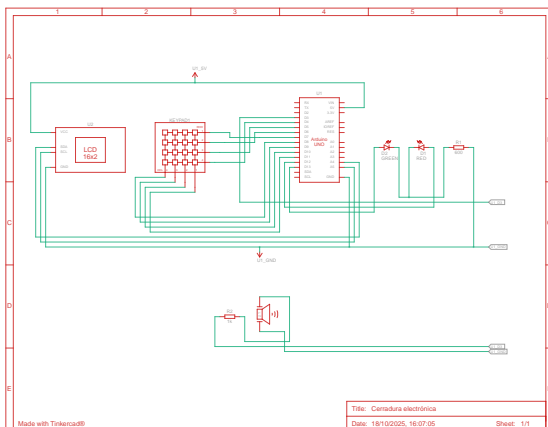


Figura 2. Esquema de conexión del candado electrónico. Elaboración propia en Tinkercad®.

VI-B. Códigos fuente

VI-B1. Piano electrónico: A continuación se detalla la estructura de archivos del proyecto del piano electrónico:

```
-- main.c
-- piano.h
-- piano.c
-- sonidos.h
-- sonidos.c
-- funciones.h
-- funciones.c
-- figuras.h
-- figuras.c
-- canciones.h
-- canciones.c
-- hw_pins.h
-- hw_pins.c
-- uart.h
-- uart.c
```

El código fuente completo del proyecto se encuentra disponible en el repositorio de GitHub [7].

VI-C. Evidencias

VI-D. Tablas complementarias