

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.

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.

I. INTRODUCCIÓN

Objetivos específicos

II. MARCO TEÓRICO

II-A. Piano electrónico con ATmega328P

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

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. Estado de avance: Hasta el momento, se han definido los pines correspondientes a las 12 notas musicales y los dos pulsadores de control de octavas. También se determinó la utilización de resistencias pull-up internas para simplificar el diseño del circuito. El siguiente paso consiste en implementar la generación de tonos mediante PWM y la integración de la comunicación serial para las canciones predefinidas.

III-E. Cerradura electrónica

IV. RESULTADOS

IV-A. Plotter

IV-B. Seleccionador de colores

IV-C. Piano

IV-D. Cerradura electrónica

V. CONCLUSIONES

V-A. Plotter

V-B. Seleccionador de colores

V-C. Piano

V-D. Cerradura electrónica

REFERENCIAS

- [1] Microchip Technology Inc. Atmega328p datasheet. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- [2] circuito.io. (2018) Arduino uno pinout diagram. [Online]. Available: <https://www.circuito.io/blog/arduino-uno-pinout/>
- [3] ARXterra. Addressing modes — 8-bit avr instruction set. [Online]. Available: <https://www.arxterra.com/3-addressing-modes/>
- [4] Software Particles. (2024, Apr.) Learn how an 8x8 led display works and how to control it using an arduino. Figura: 8x8 LED matrix pin mapping (imagen del artículo). [Online]. Available: <https://softwareparticles.com/learn-how-an-8x8-led-display-works-and-how-to-control-it-using-an-arduino/>
- [5] Carpeta del laboratorio (google drive). Carpeta compartida del laboratorio. [Online]. Available: <https://drive.google.com/drive/u/0/folders/1fP0aLlozXeapRgDPDNWT1TrhAYr1PPPT>
- [6] Microchip Community (AVR Freaks). Avr freaks — comunidad de desarrolladores avr. Foros técnicos y soluciones prácticas sobre AVR. [Online]. Available: <https://www.avrfreaks.net/>
- [7] J. Ganssle. A guide to debouncing. Referencia clásica para anti-rebote en pulsadores/encoders. [Online]. Available: <https://www.ganssle.com/debouncing.htm>
- [8] G. Schmidt. (2021, Sep.) Beginners introduction to the assembly language of atmel-avr-microprocessors. Tutorial de avr-asm-tutorial.net (revisión septiembre 2021). [Online]. Available: <https://kitsandparts.com/tutorials/assemblers/BeginnersAVRasm.pdf>
- [9] T. Redelberger. (2019, Apr.) Avr assembler for complex projects. Versión 0.4 (2019-04-06). [Online]. Available: https://web222.webclient5.de/doc/swdev/avrasm/advavrasm2/AdvancedUseAVRASM2_en_20190406.pdf
- [10] Laboratorio de Microcontroladores, UTEC, “Repositorio de laboratorio de microcontroladores (tec.micro),” <https://github.com/MateoLecuna/Tec.Micro>, 2025, accedido el 26 de septiembre de 2025.

VI. ANEXOS