

Desarrollo de un equipo para enviar secuencias MIDI programables

Gutiérrez, Lucas¹, Volentini, Esteban D.², Juarez, Martín G.²

¹ Ing. en Computación, Facultad de Ciencias Exactas y Tecnología, Universidad Nacional de Tucumán.

² Depto. de Electricidad, Electrónica y Computación, Facultad de Ciencias Exactas y Tecnología, Universidad Nacional de Tucumán

Resumen

MIDI (Musical Instrument Digital Interface) es un estándar desarrollado para la comunicación entre instrumentos musicales digitales, computadoras y otros dispositivos de audio. Los mensajes MIDI transmiten información sobre las notas que debe ejecutar el generador de sonido, por lo que, sin este último, no se produce ningún sonido. La especificación contempla 16 canales de transmisión, 120 notas posibles y una polifonía (cantidad de notas que pueden sonar simultáneamente) que depende del protocolo de comunicación utilizado, como UART o USB.

Este proyecto consiste en la implementación de un secuenciador MIDI, maximizando los parámetros del estándar, mediante el uso de un microcontrolador RP 2040. El dispositivo funciona de manera autónoma y puede conectarse tanto a un sintetizador compatible con MIDI como a una computadora para su uso con estaciones de trabajo de audio digital (DAWs), a través de los conectores adecuados. Además, puede operar en modos de comunicación full-dúplex o simplex, según se utilice un reloj MIDI externo o el reloj interno del sistema.

El secuenciador cuenta con distintos modos de funcionamiento:

- Modo Secuenciador o Programación: permite crear, guardar, cargar y reproducir secuencias MIDI de hasta 32 pasos.
- Modo Play: reproduce las secuencias almacenadas, permitiendo cambiar entre ellas rápidamente mediante bancos de secuencias.

Las secuencias se almacenan como archivos MIDI en una tarjeta SD y pueden ser transferidas o modificadas desde una computadora. El dispositivo incluye 16 botones para seleccionar los pasos de la secuencia, acompañados por 16 LEDs RGB para su visualización, además de otros botones para controlar funciones adicionales. También incorpora una pantalla que muestra información sobre las secuencias y sus parámetros.

Durante el prototipado se utilizó el motor de videojuegos Godot, simulando los dispositivos de entrada y salida. Esto permitió desarrollar la lógica del sistema en GDScript, un lenguaje de alto nivel, utilizando herramientas gráficas simples e intuitivas, y generar un ejecutable fácilmente distribuible para pruebas de funcionamiento y experiencia de usuario.

La implementación final se realizó en C++ utilizando el sistema operativo en tiempo real Mbed CE (una continuación comunitaria de Mbed OS). Se aplicó un modelo de composición de clases, dividiendo el sistema en módulos como control de botones, pantalla, gestión del protocolo MIDI, entre otros.

El resultado es un dispositivo de bajo costo, fácil de usar, útil para músicos en vivo y valioso como experiencia de aprendizaje en sistemas embebidos y programación de bajo nivel con RTOS.

Introducción

El alumno, con experiencia en el ámbito musical, identificó una necesidad común entre músicos amateur: la dificultad de acceder a equipos MIDI de alta gama debido a sus elevados costos. Esta problemática, frecuente en contextos de producción musical independiente, motivó el desarrollo de un dispositivo MIDI autónomo, de bajo costo y accesible, que conserve las funcionalidades principales de los equipos profesionales.

En el contexto actual, donde la tecnología digital ocupa un rol central en la creación musical, los dispositivos MIDI son herramientas fundamentales para la composición, ejecución y experimentación sonora. Sin embargo, muchas veces su adquisición está limitada a músicos con mayores recursos. Esto abre una oportunidad para el desarrollo de soluciones tecnológicas que, aprovechando microcontroladores de bajo costo y libre disponibilidad, puedan replicar estas funciones en forma eficiente.

La hipótesis de trabajo plantea que es posible implementar un secuenciador MIDI autónomo, flexible y de bajo costo utilizando un microcontrolador RP 2040, integrando hardware de fácil acceso y software abierto, sin comprometer la funcionalidad esperada en entornos musicales reales. Se propone que, con una correcta arquitectura del sistema y un diseño centrado en la experiencia del usuario, se puede alcanzar una herramienta útil tanto para músicos en vivo como para producción en estudio.

El presente trabajo se enfoca entonces en el diseño e implementación de un secuenciador MIDI utilizando el microcontrolador RP 2040, programado en C++ sobre el sistema operativo en tiempo real Mbed CE. El dispositivo permite crear, almacenar y reproducir secuencias MIDI de hasta 32 pasos, operando de forma autónoma o en conjunto con sintetizadores o computadoras mediante conexiones estándar MIDI. Además, se prevé una interfaz intuitiva mediante botones, LEDs RGB y pantalla, junto con almacenamiento en tarjeta SD.

La propuesta no solo busca resolver una necesidad personal del alumno como músico, sino también explorar e integrar conceptos de programación de sistemas embebidos, protocolos de comunicación digital y diseño de interfaces, aportando así una experiencia formativa completa y con aplicación práctica en el campo de la música y la electrónica.

Objetivo General

Diseñar e implementar un dispositivo de bajo costo para realizar las funciones de secuenciador MIDI multi-canal.

Objetivos Específicos

- Contar con una interfaz de usuario intuitiva, validada por músicos, mostrada en una pantalla LCD.
- Aprovechar el protocolo MIDI en su totalidad, utilizando al máximo posible sus capacidades.
- Permitir la conexión a través de UART (utilizando los parámetros del estándar MIDI) o USB (como un Dispositivo USBMIDI) a un sintetizador o una computadora.
- Proveer la capacidad de dividir el tiempo musical en hasta 32 pasos.
- Disponer de por lo menos 16 botones físicos para representar cada paso de la secuencia, y si es necesario, un botón adicional para cambiar entre los primeros o los últimos pasos.
- Realizar un prototipo para probar las funcionalidades del dispositivo.

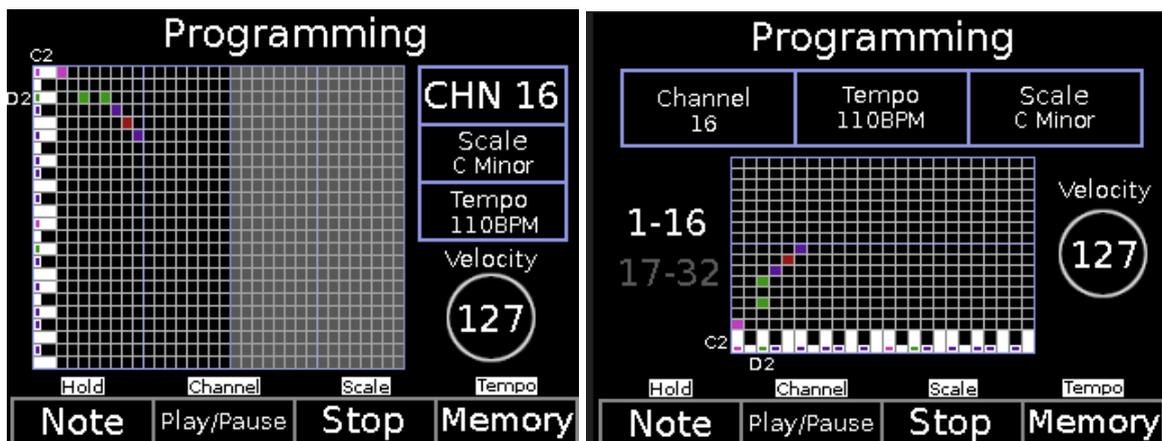
Estado del Arte

En el mercado existen secuenciadores MIDI comerciales de distintas gamas. Equipos como el [Elektron Digitakt](#), el [Arturia BeatStep Pro](#) o el [Novation Circuit](#) ofrecen soluciones compactas, con múltiples funciones, interfaces amigables y alta calidad de construcción. Sin embargo, su precio suele estar fuera del alcance de muchos usuarios, especialmente estudiantes, aficionados o músicos independientes. Esto ha incentivado la aparición de alternativas DIY (Do-It-Yourself), muchas de ellas basadas en plataformas como Arduino, Teensy o Raspberry Pi, las cuales permiten implementar dispositivos MIDI personalizados mediante programación.

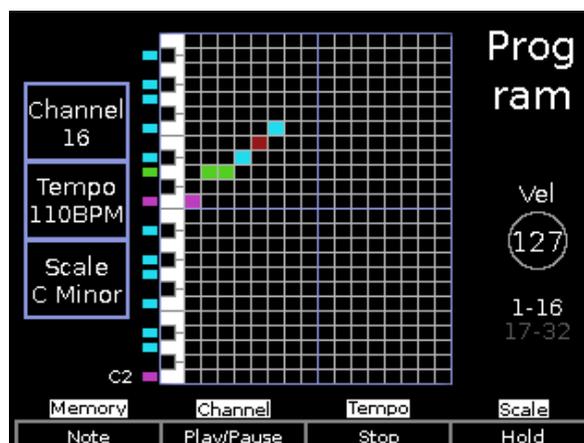
Existen ejemplos comunitarios de implementación de secuenciadores MIDI con RP2040, pero en su mayoría están limitados a funciones básicas, están especializados para creación generativa de música, o carecen de interfaces completas de usuario. Algunos de estos son [Calculator](#), [picostepseq](#) o [Pico-Sequencer](#).

Desarrollo

Primero, se diseñó la interfaz utilizando GIMP y la técnica de Pixel Art, debido a que la pantalla utilizada tiene una resolución de 320x240px, por lo que la interfaz se debe ajustar a estas medidas. El primer diseño utilizaba una visualización horizontal de los pasos (es decir, hacia la derecha era el tiempo superior), mostrando los 32 pasos. El segundo, cambió a una visualización vertical, pero solo mostrando 16 pasos, permitiendo cambiar los pasos mostrados con un botón.

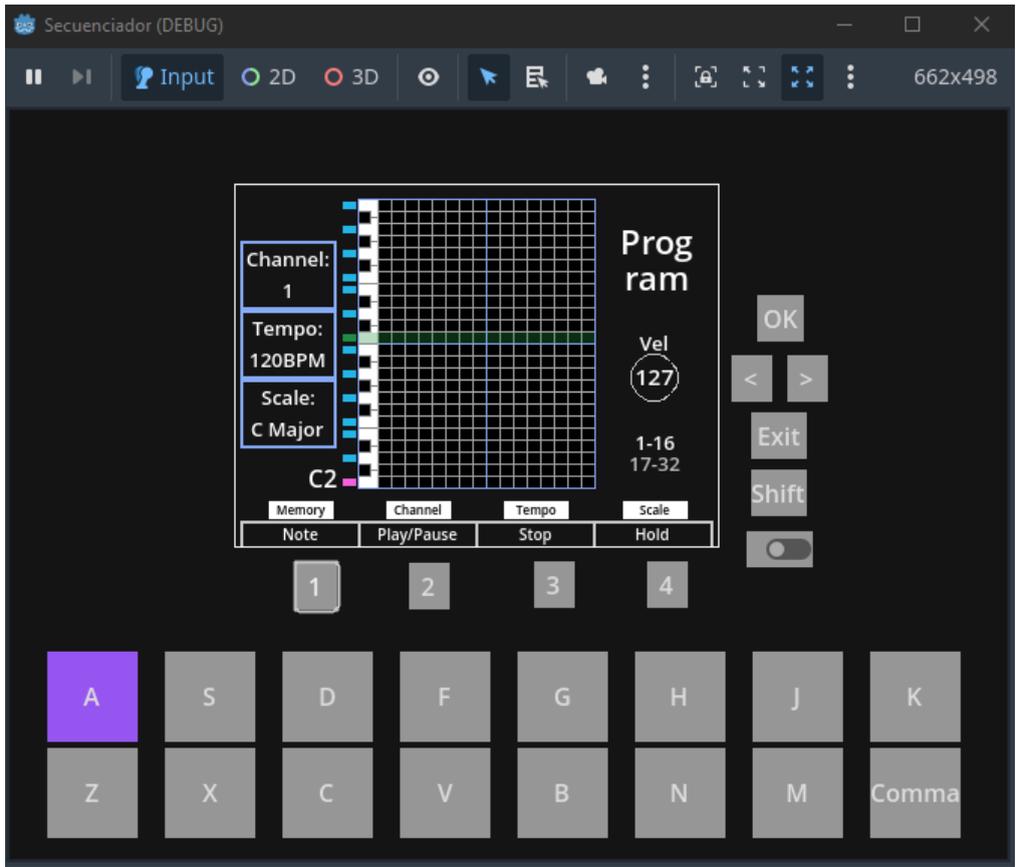


Se implementaron estas interfaces en Godot para probar su funcionamiento, y, luego de unos cambios pequeños, se determinó que la visualización horizontal con 16 pasos era la mejor opción, por lo que la interfaz final queda como la siguiente imagen.

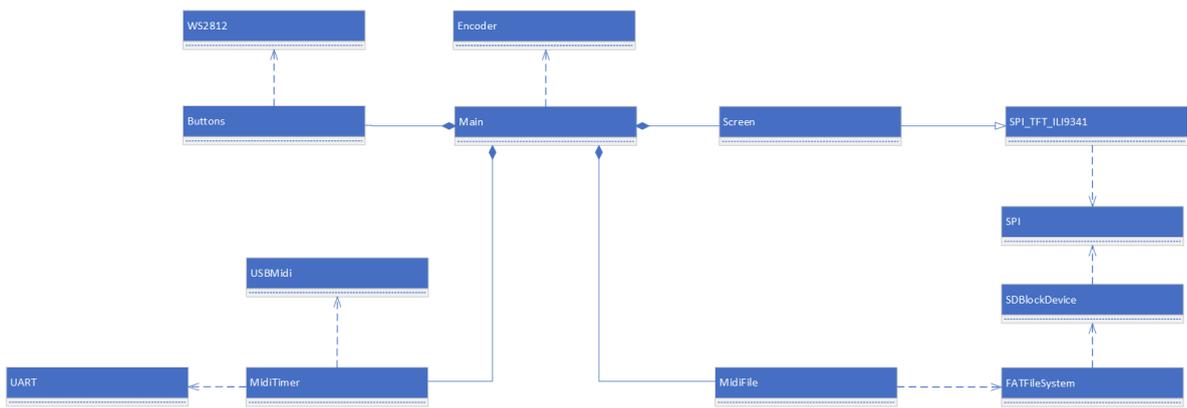


A partir de esta interfaz, se [desarrolló en Godot](#) toda la lógica del equipo, incluyendo agregado y borrado de notas, cambios de parámetros como canal MIDI, Tempo, Escala, Velocidad, además de la funcionalidad de guardar y leer archivos MIDI estándar.

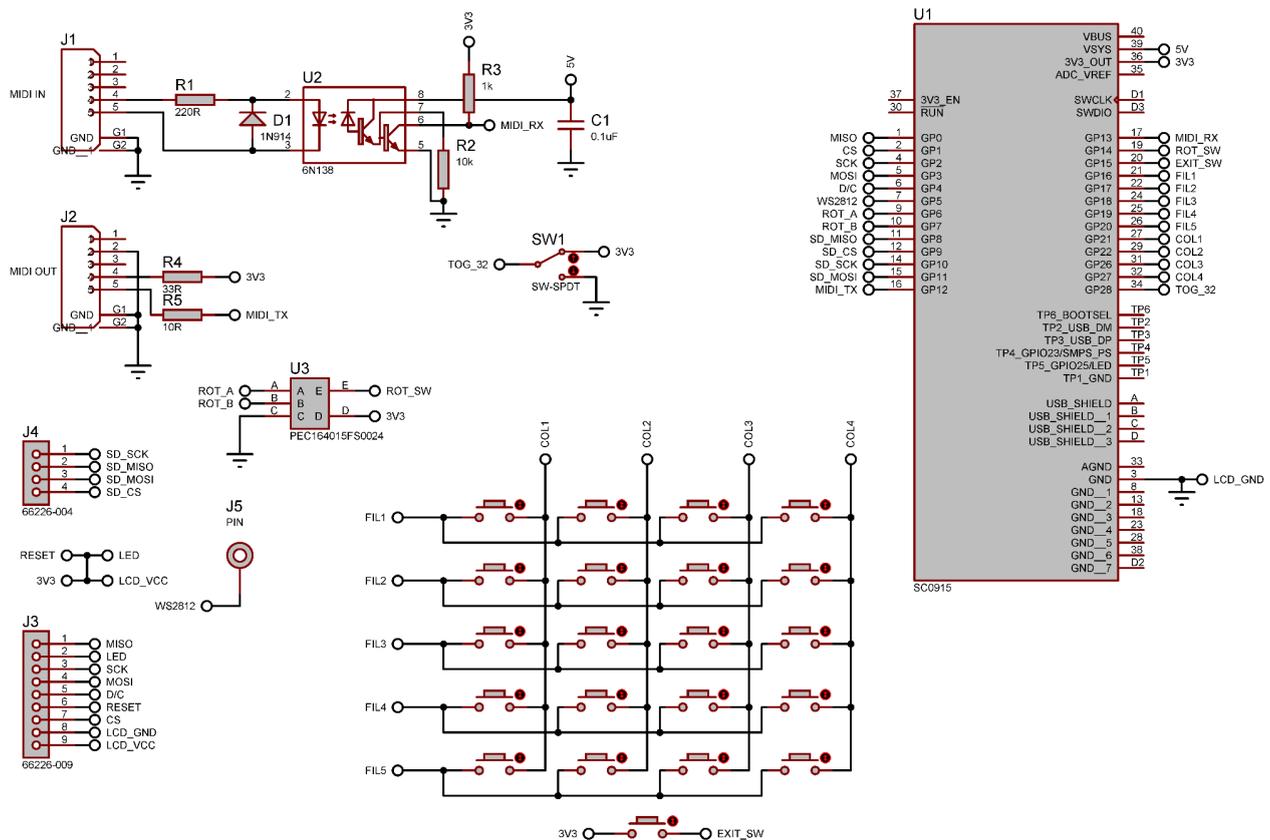
También se agregaron los dispositivos de interacción humana, incluyendo 16 botones para los pasos, 4 botones de función, más un encoder rotatorio con switch (representado por dos flechas izquierda y derecha, más un botón), un botón para salir (y uno de shift, el cual en el dispositivo físico será mantener el botón de salir) y un toggle para elegir si se manejan 32 o 16 pasos. Para la utilización de estos botones, se mapeó al teclado de la computadora cada botón.



Con toda la lógica del sistema realizada, se realizó un diagrama de clases utilizando el modelo de composición, para luego implementar el mismo en el sistema embebido. Este diagrama debe incluir además los drivers (proporcionados por MBed CE) para poder interactuar con los periféricos conectados al microcontrolador, entre ellos la pantalla TFT conectada por SPI, la tarjeta SD conectada por SPI, el encoder rotatorio, los LEDs WS2812 y los botones



Además, se realizó un diagrama de conexión al microcontrolador utilizando Proteus, para realizar los circuitos necesarios.



Con los diagramas finalizados, se realizó la [implementación en C++, con el RTOS MBed CE](#). Para cargar el código en el microcontrolador, se utiliza el bootloader integrado de las placas de desarrollo Raspberry Pi Pico, y la herramienta Picotool. Para el manejo de los LEDs WS2812, los cuales solo necesitan un pin de datos para los 16 LEDs, se utiliza la funcionalidad PIO del RP 2040, utilizando la máquina de estados interna como generador para los pulsos correspondientes.

Experimentación

Luego de la implementación, se fue probando módulo a módulo la funcionalidad del dispositivo, resolviendo errores y modificando la interfaz para una mejor visualización. Los errores encontrados fueron:

- Manejo de la pantalla a partir de las librerías disponibles para el RTOS.
- Uso de USB, principalmente la enumeración como dispositivo USBMIDI o USBMSD.
- Diferencias de sintaxis entre GDScript y C++.
- Implementación de SD con SPI.

Todos estos errores fueron solucionados en distintas iteraciones, exceptuando la implementación de la tarjeta SD, que no pudo ser resuelta, por lo que se utilizó un FileSystem basado en Heap, dejando para el futuro la implementación del sistema final de archivos.

Conclusiones y trabajos futuros

El desarrollo del secuenciador MIDI autónomo cumplió con el objetivo general propuesto: diseñar e implementar un dispositivo de bajo costo con capacidades multi-canal, orientado a músicos amateur o estudiantes. Se alcanzaron la mayoría de los objetivos específicos: se logró una interfaz de usuario intuitiva con retroalimentación visual mediante LEDs RGB, se utilizaron plenamente las capacidades del protocolo MIDI (canalización, mensajes estándar, almacenamiento), y se permitió la conexión tanto por UART como por USB.

Además, se incorporó un flujo de trabajo iterativo que permitió validar funcionalidad a través de simulaciones en Godot, lo que facilitó el diseño lógico antes de la implementación embebida. Esta etapa resultó especialmente valiosa para depurar la lógica de funcionamiento, validar la usabilidad y estructurar la arquitectura de clases que luego fue trasladada al entorno en tiempo real (RTOS).

Entre las limitaciones actuales, se encuentra la longitud máxima de 32 pasos por secuencia, que podría ampliarse mediante una interfaz de usuario más dinámica. Otra limitación está en la polifonía y profundidad del control MIDI: si bien se usan canales múltiples, no se implementaron aún cambios de programa o control de expresión más complejos. La interfaz física podría mejorarse integrando encoders de mayor resolución o pantallas de mayor densidad, así como botones retroiluminados para entornos oscuros.

Como líneas de trabajo futuras, se propone:

- Permitir capturar y grabar secuencias desde otro controlador.
- Más modos de funcionamiento, permitiendo manejar DAWs y sintetizadores en tiempo real.
- Crear una carcasa impresa en 3D, o en metal para más rigidez.
- Crear un PCB dedicado para integrar el hardware de forma definitiva.
- Implementar FileSystem con Tarjeta SD.

Referencias Bibliográficas

1. Raspberry Pi Foundation. "RP2040 Datasheet." <https://datasheets.raspberrypi.com/rp2040>
2. Adafruit. *WS2812 Datasheet*. <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>
3. Adafruit. *ILI9341 Datasheet*. <https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf>
4. Elektron. *Digitakt II – Digital Drum Computer & Sampler*. <https://www.elektron.se>
5. Arturia. *BeatStep Pro – Controller & Sequencer*. <https://www.arturia.com/products/hybrid-synths/beatstep-pro/overview>
6. Novation. *Circuit Tracks*. <https://novationmusic.com/products/circuit-tracks>
7. Todbot. *PicoStepSeq*. <https://github.com/todbot/picostepseq>
8. Rheslip. *Pico-Sequencer*. <https://github.com/rheslip/Pico-Sequencer>
9. P-o-l-e. *Calculator – RP2040 MIDI Sequencer*. <https://github.com/p-o-l-e/calculator>
10. MBed CE Project. <https://github.com/MBedCE>
11. Lucas Gutierrez. *Simulador en Godot*. <https://github.com/LucastuFett/GodotSecuenciador>
12. Lucas Gutierrez. *Implementación en Mbed CE*. <https://github.com/LucastuFett/SecuenciadorMBed>