SP1 Mk2 (ESP32-S3) — Documentación Técnica Ampliada

# 1. Introducción General El SP1 Mk2 es un controlador MIDI modular avanzado, desarrollado sobre un microcontrolador ESP32-S3. Está diseñado como núcleo lógico del sistema SP1, gestionando pantallas OLED, navegación por menús, presets en tarjeta SD, un secuenciador de pasos editable y sincronización externa por MIDI Clock (DIN y USB). Descripción del Firmware:

El firmware para el controlador MIDI SP1 Mk2 es un sistema embebido avanzado que combina funcionalidades de superficie de control MIDI, teclado MIDI y secuenciador paso a paso. Diseñado para ejecutarse en placas ESP32-S3, este firmware ofrece una interfaz de usuario intuitiva mediante una pantalla OLED y un encoder rotativo para navegación.

## Características Principales

### 1. ****Sistema de Menú Unificado****

* Interfaz estructurada en 3 zonas interactivas:
  + **Zona Superior**: Menús principales y submenús
  + **Zona Central**: Parámetros de grupo y controles
  + **Zona Inferior**: Valores y acciones
* Navegación mediante encoder rotativo (giro y pulsación)
* Sistema de parpadeo visual para indicar selección activa
* Timeout automático para regresar a pantalla principal

### 2. ****Secuenciador Paso a Paso****

* 4 pistas independientes con hasta 64 pasos configurables
* Parámetros editables por paso:
  + Activación (On/Off)
  + Nota musical (0-127)
  + Velocidad (0-127)
  + Duración (en ticks)
* Configuración global:
  + Tempo (BPM) ajustable (30-300)
  + Sincronización MIDI externa (MIDI Clock)
  + Modo Mono/Poli
  + Ajustes de Legato y Sustain
* Reproducción con manejo inteligente de notas activas

### 3. ****Gestión de Controles MIDI****

* 108 controles físicos organizados en grupos
* Configuración completa por control:
  + Nombre personalizado
  + Canal MIDI (0-15)
  + Número CC (0-127)
  + Valor actual (0-127)
* Identificación mediante multiplexores (MUX 0-8)

### 4. ****Sistema de Presets****

* Almacenamiento en tarjeta SD (formato CSV)
* Gestión independiente para:
  + Presets de Superficie (controladores)
  + Presets de Secuenciador
* Carga/guardado de configuración completa

### 5. ****Comunicación MIDI****

* Soporte dual:
  + **USB MIDI** mediante TinyUSB
  + **UART MIDI** tradicional (31250 baudios)
* MIDI Thru configurable
* Sincronización con reloj MIDI externo (MIDI Clock)

### 6. ****Sistema de Pantallas Inteligente****

* Transiciones dinámicas entre modos:
  + **Pantalla Home**: Estado del sistema
  + **Pantalla Standby**: Muestra eventos MIDI en tiempo real
  + **Pantalla Menú**: Navegación completa
* Actualización por zonas para optimización
* Animación de inicio personalizada

### 7. ****Hardware Específico****

* Pantalla OLED 128x128 (SSD1351)
* Entrada encoder rotativo (A/B/Botón)
* Soporte para tarjeta SD
* 9 multiplexores (74HC4067) para 108 controles

## Estructura del Código

### Módulos Principales:

1. **controles(.h/.cpp)**

* Gestión central de todos los controles físicos
* Configuración MIDI por control
* Funciones de carga/guardado de presets

1. **secuenciador(.h/.cpp)**

* Motor del secuenciador paso a paso
* Programación de pasos y patrones
* Sincronización MIDI

1. **menu\_unico(.h/.cpp)**

* Lógica completa del sistema de menús
* Navegación jerárquica
* Vinculación con controles físicos

1. **pantalla\_navegacion(.h/.cpp)**

* Máquina de estados de pantallas
* Gestión de transiciones y timeout
* Visualización de eventos MIDI

1. **pantalla\_unica(.h/.cpp)**

* Renderizado de la interfaz de usuario
* Sistema de dibujado optimizado

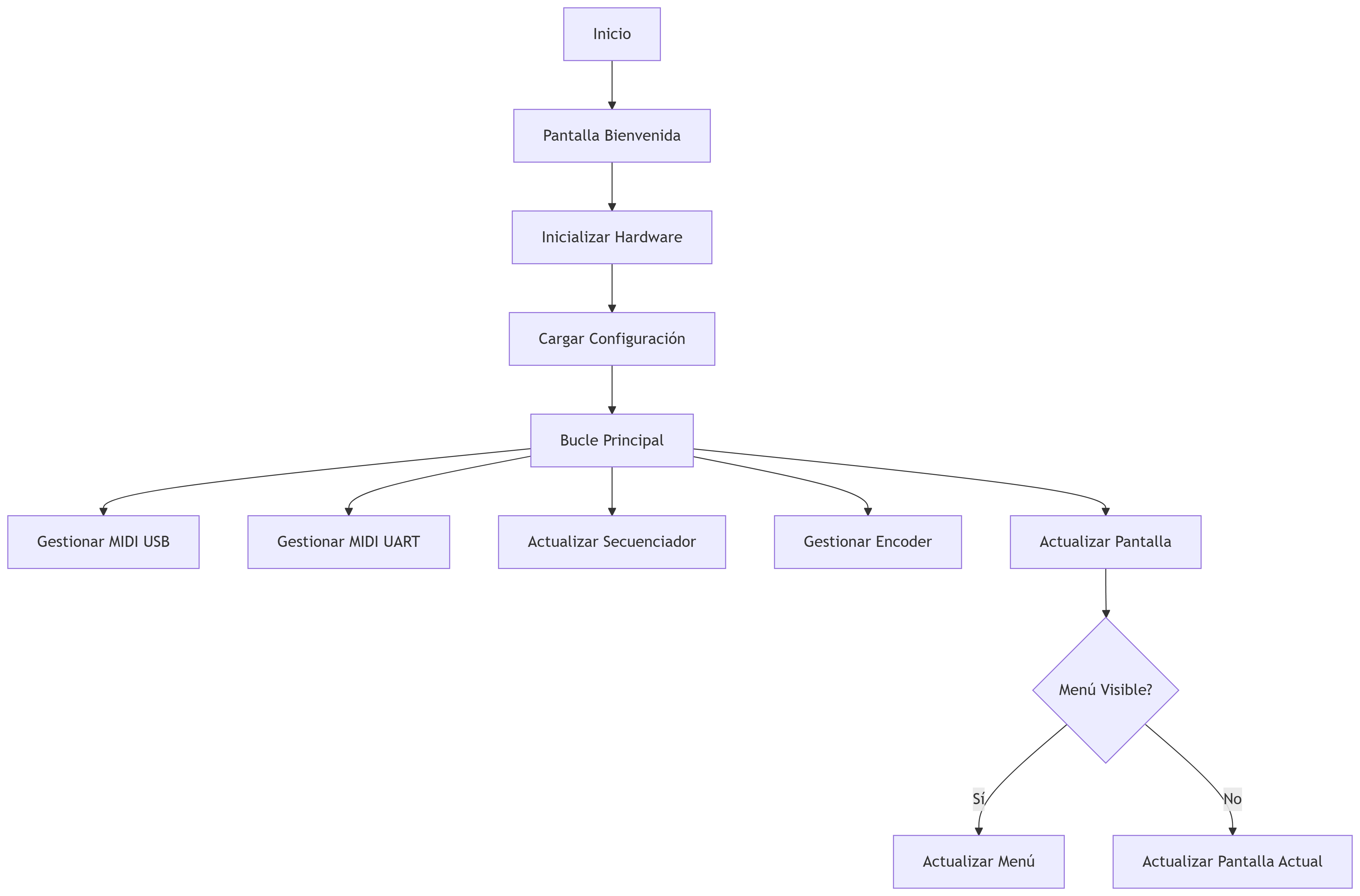
1. **configuracion(.h/.cpp)**

* Parámetros globales del sistema
* Variables de estado persistentes

1. **zona\_menu(.h/.cpp)**

* Definición de zonas interactivas
* Validación de estructura de menús

### Flujo Principal:



## Instalación y Uso

### Requisitos:

* Placa ESP32-S3
* Librerías:
  + Adafruit\_SSD1351
  + Adafruit\_TinyUSB
  + SPI
  + SD

### Configuración Hardware:

cpp

*// hardware\_config.h*

#define OLED\_CS 5

#define OLED\_DC 15

#define OLED\_RST 16

#define ENCODER\_A\_PIN 9

#define ENCODER\_B\_PIN 10

#define ENCODER\_BUTTON\_PIN 13

#define SD\_CS 8

### Funcionamiento Básico:

1. **Navegación Principal**:
   * Giro del encoder: Cambiar valores/selecciones
   * Pulsación corta: Confirmar selección
   * Pulsación larga: Cambiar entre zonas
2. **Modo Secuenciador**:
   * Edición paso por paso
   * Configuración de patrones
   * Sincronización con reloj externo
3. **Gestión de Presets**:
   * Cargar configuraciones predefinidas
   * Guardar configuraciones personalizadas
   * Gestión por archivos CSV en SD

## Cambios Relevantes en v0.6.1

### Nuevas Funcionalidades:

1. **Sincronización MIDI Externa**:
   * Soporte completo para MIDI Clock
   * Configuración BPM Sync en menú
   * Detección de ticks de reloj
2. **Sistema de Encoder Mejorado**:
   * Máquina de estados para detección precisa
   * Debounce mejorado (50ms)
   * Soporte para giro continuo
3. **Optimización de Pantalla**:
   * Actualización por zonas
   * Reducción de flickering
   * Texto centrado dinámico
4. **Gestión de Notas Activas**:
   * Sistema de apagado automático
   * Control de duración preciso
   * Visualización en pantalla

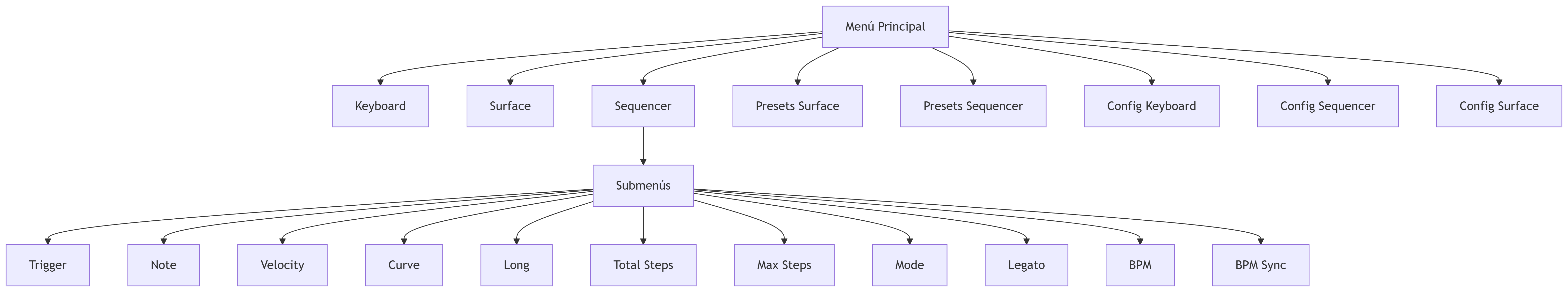
### Correcciones:

* Solucionado redibujado innecesario en eventos MIDI
* Corregida inicialización de pasos del secuenciador
* Normalización de nombres de variables
* Optimización de consumo de memoria

### Problemas Conocidos:

* Límite de 108 controles físicos
* Retardo en carga de presets grandes
* Limitaciones en visualización de texto largo

## Diagrama de Menú Principal



## Instrucciones de Compilación

1. Instalar ESP32 Arduino Core
2. Importar librerías requeridas
3. Configurar placa como "ESP32S3 Dev Module"
4. Habilitar USB CDC On Boot
5. Compilar y cargar firmware

# 2. Arquitectura del Sistema

El sistema está compuesto por dos módulos principales: el SP1 Mk2 (ESP32-S3) como cerebro central, y el SP1 Mk1 (Arduino Micro) como esclavo multiplexador que gestiona los 108 controles físicos.  
La comunicación se realiza vía UART a 31250 bps, y el Mk2 también actúa como dispositivo USB MIDI estándar.

Flujo general:  
- SP1 Mk1 → Envía cambios de control al Mk2 vía UART (#MUX, #CC, #VAL)  
- SP1 Mk2 → Interpreta los datos, actualiza interfaz OLED, y reenvía por USB MIDI  
- Usuario → Navega menús, ajusta parámetros, guarda/carga presets desde SD

# 3. Menús y Navegación OLED

El sistema visual está dividido en 3 zonas: superior, central e inferior. Cada pantalla OLED tiene 5 líneas configurables por menú. La navegación se realiza con un encoder rotatorio con clic.

Pantallas implementadas:  
- Inicio / Test / Standby (transición automática)  
- CONTROL (último control recibido)  
- PRESETS: LOAD / SAVE / DELETE  
- CONFIG: ajustes globales como MIDI Thru, Sync, Mute, etc.  
- SEQUENCER: edición de pasos, tempo, longitud, modo mono, canal

# 4. Funcionalidad del Secuenciador

El módulo secuenciador gestiona hasta 4 pistas paralelas, con matrices de pasos configurables:  
  
- `Step secuencia[NUM\_SEQUENCERS][MAX\_STEPS\_CONFIGURABLE]`  
Cada `Step` contiene: activo, nota, velocidad, inicio, duración.  
  
La reproducción se realiza por `avanzarPaso()` o `avanzarSecuencia()` y permite muteo, legato, sustain, modo mono y polifónico. Se controlan notas reales por USB y UART MIDI.

# 5. Sincronización Externa MIDI Clock

El firmware detecta mensajes `0xF8` de MIDI Clock por USB y UART. Cuando está activo `bpmSyncEnabled`, los pasos avanzan en función de `midiClockTicks`, que se incrementa con cada `0xF8`.   
El número de ticks necesarios por paso (`ticksPorStep`) es actualmente fijo, pero puede ampliarse vía menú.  
  
Se planea compatibilidad futura con `0xFA`, `0xFB` y `0xFC` para integración con DAWs como Ableton, Bitwig y FL Studio.

# 6. Presets y Almacenamiento SD

Dos carpetas gestionan los presets:  
- `/Presets Surface/` → Mapa físico de controles (cc, canal, valor)  
- `/Presets Seq/` → Contenido completo del secuenciador por pasos  
Los archivos `.csv` permiten edición externa y carga automática al inicio.  
  
Desde el menú PRESETS, se permite: seleccionar acción (LOAD / SAVE / DELETE), elegir nombre de preset, y confirmar (OK / CANCEL).

# 7. Variables Globales

- `estadoTempo` → BPM manual cuando no se sincroniza por MIDI Clock  
- `bpmSyncEnabled` → Flag de activación de modo clock externo  
- `midiClockTicks` → Contador de 0xF8  
- `muteSequencerNotes` → Silencia el secuenciador sin apagar su lógica  
- `secuenciaTecladoLinkeada` → Dispara secuencia con teclado  
- `porcentajeLegato`, `porcentajeSustain` → Afectan duración de las notas  
- `secuenciadorGlobalActivo` → Pausar/reanudar secuencia

# 8. Componentes de Hardware y Pines

| Nombre lógico | GPIO | Función |  
|--------------------|------|--------------------------|  
| MIDI\_UART\_TX | 14 | Salida MIDI DIN |  
| MIDI\_UART\_RX | 13 | Entrada MIDI DIN |  
| ENCODER\_A\_PIN | 34 | Encoder menú A |  
| ENCODER\_B\_PIN | 35 | Encoder menú B |  
| ENCODER\_BUTTON\_PIN | 36 | Botón encoder |  
| OLED\_CS | 5 | Chip select OLED |  
| OLED\_DC | 16 | OLED D/C |  
| OLED\_RST | 17 | OLED Reset |  
| SD\_CS | 10 | Chip select tarjeta SD |

# 9. Estructura Modular del Código

- `FIRMWARE\_SP1\_Mk2\_ESP32S3.ino`: ciclo principal, llamadas globales  
- `hardware\_config.h`: definición de pines y constantes  
- `pantalla\_inicio.\*`: animación de arranque  
- `pantalla\_unica.\*`: interfaz principal OLED  
- `pantalla\_navegacion.\*`: transiciones y navegación  
- `menu\_unico.\*`: lógica de menús por zonas  
- `secuenciador.\*`: reproducción, sincronía, presets  
- `configuracion.\*`: variables globales y EEPROM  
- `controles.\*`: gestión física, comunicación UART

# 10. SISTEMA GENERAL

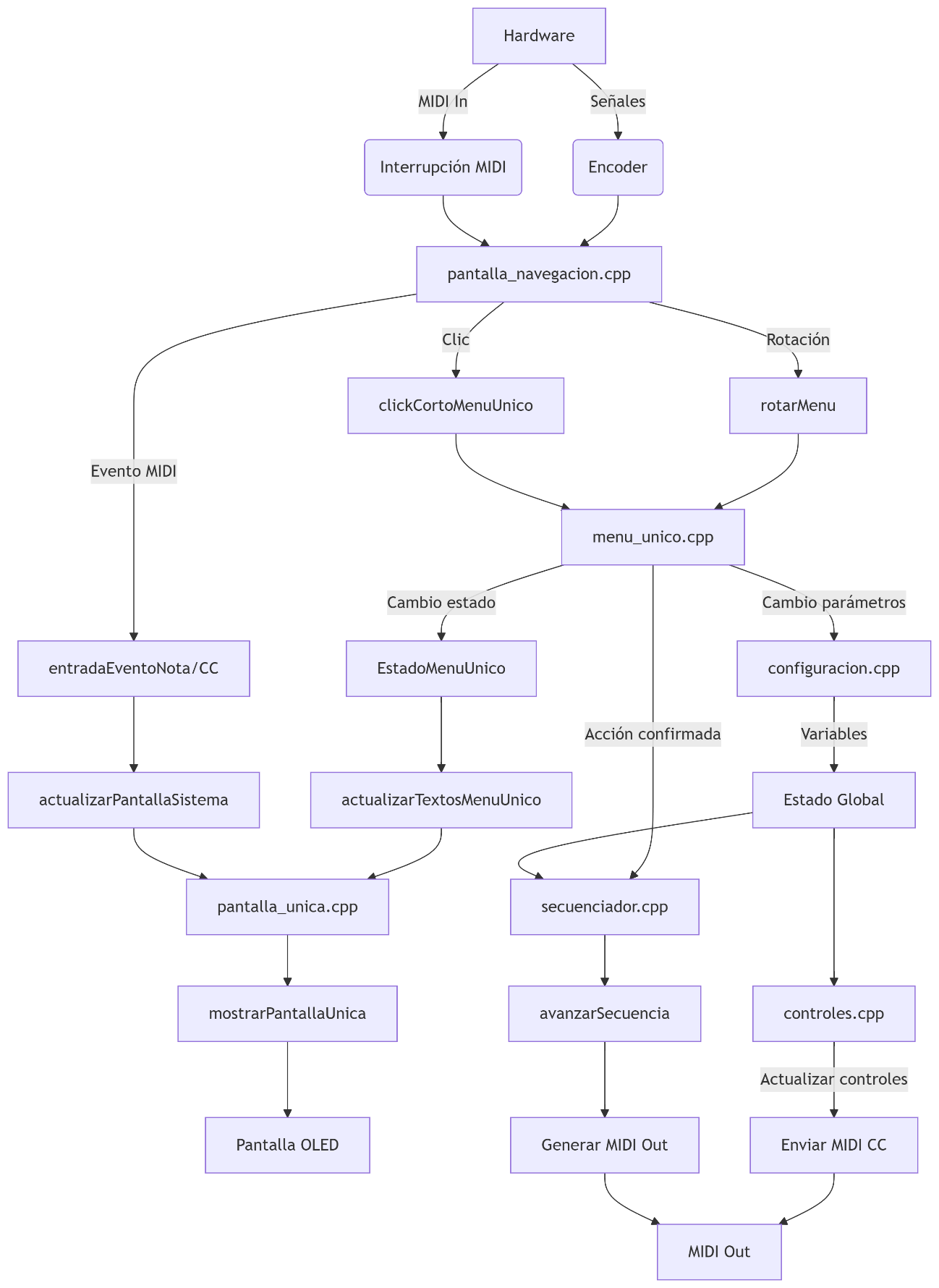
**Componentes Principales:**

1. **Gestión de Menú** (menu\_unico, zona\_menu)
2. **Control de Pantalla** (pantalla\_navegacion, pantalla\_unica)
3. **Secuenciador MIDI** (secuenciador)
4. **Gestión de Controles** (controles)
5. **Configuración Global** (configuracion)
6. **Hardware** (Encoder, Pantalla OLED, Entradas MIDI)

**Flujos Clave:**

1. Flujo de interacción del usuario (encoder)
2. Flujo de eventos MIDI
3. Flujo de actualización de pantalla
4. Flujo del secuenciador

Diagrama de Flujo Detallado:



Módulos y Responsabilidades:

1. pantalla\_navegacion.cpp (Control central):
   * Gestión de eventos del encoder
   * Recepción de eventos MIDI
   * Control de timeout de pantalla
   * Transición entre pantallas
2. menu\_unico.cpp (Lógica de menú):
   * Gestión de estados del menú
   * Navegación por zonas
   * Confirmación de acciones
   * Actualización de textos
3. pantalla\_unica.cpp (Renderizado):
   * Dibujado de elementos en pantalla
   * Manejo de colores y zonas
   * Actualización parcial/full de display
4. secuenciador.cpp (Núcleo musical):
   * Avance de secuencias
   * Generación de eventos MIDI
   * Gestión de tempo y sincronización
   * Manejo de pasos y patrones
5. controles.cpp (Interfaz física):
   * Mapeo de controles a CC MIDI
   * Gestión de grupos y subgrupos
   * Actualización de valores
6. configuracion.cpp (Estado global):
   * Variables compartidas
   * Presets y configuraciones
   * Flags de estado del sistema

Flujos de Comunicación:

1. **Usuario → Sistema**:
   * Rotación encoder → Cambio valores/selección
   * Clic encoder → Confirmación/navegación
2. **MIDI In → Sistema**:
   * Notas → Actualización pantalla standby
   * CC → Actualización valores controles
3. **Sistema → MIDI Out**:
   * Secuenciador → Notas programadas
   * Controles → Valores CC en tiempo real
4. **Módulos Internos**:
   * Menú ⇄ Configuración (lectura/escritura valores)
   * Pantalla ⇄ Menú (estado actual)
   * Secuenciador ⇄ Configuración (parámetros)

# 11. Futuras Ampliaciones

- Guardado de BPM, mute, mono, canal en preset `.csv`  
  
- Sincronización visual con DAW en OLED (barras, compás)  
  
- Soporte para `START`, `STOP`, `CONTINUE` por MIDI Clock  
  
- Presets globales de configuración