

Proto-Synth v2: Guía de Experimentación y Programación con IA

Bienvenido al **Proto-Synth v2** de GC Lab Chile. Esta placa no es solo un sintetizador; es un laboratorio de desarrollo abierto basado en el potente **ESP32**. Aquí podrás crear samplers, sintetizadores granulares, controladores MIDI gestuales y más.

1. ¿Cómo programar el Proto-Synth? (El Método "Prompting Visual")

No necesitas ser un experto en C++ para empezar. Hemos diseñado el Proto-Synth v2 para que puedas programarlo con la ayuda de Inteligencia Artificial (ChatGPT, Claude, Gemini).

El superpoder es el Pinout: La imagen del mapa de conexiones (Pinout) que acompaña a este kit es tu "llave maestra".

Pasos para generar código:

1. Abre tu chat de IA preferido.
2. **Sube la imagen** del Pinout del Proto-Synth v2.
3. Escribe un "prompt" (instrucción) claro.
4. Copia el código al Arduino IDE y súbelo a la placa.

Ejemplo de Prompt Maestro:

Copia y pega esto junto con la imagen para obtener mejores resultados:





"Actúa como un experto en programación de sistemas embebidos y audio digital (DSP). Analiza la imagen del Pinout del Proto-Synth v2 adjunta.

Quiero crear un código en Arduino para el ESP32 que haga lo siguiente: [DESCRIBE TU IDEA AQUÍ: Ej. Un sintetizador de onda dientes de sierra donde el Potenciómetro 1 controle la frecuencia y el MPU6050 controle el filtro con la inclinación]

Usa la librería 'Mozzi' o salida DAC directa (Pin 25) según corresponda. Considera que los botones tienen configuración Pull-up interna. Mapea los pines exactamente como aparecen en la imagen."

2. Erratas y Consideraciones Importantes (¡Leer antes de usar!)

Como toda herramienta de hardware experimental, la versión v2 tiene sus peculiaridades que le dan carácter. Ten esto en cuenta al programar y usar la placa:

-  **Potenciómetros Invertidos:** Debido al diseño del PCB, los potenciómetros funcionan al revés (Giran a la derecha = bajan el valor).
 - *Solución:* En tu código, invierte la lectura matemáticamente. Ejemplo:
`val = 4095 - analogRead(pin);`
 -  **WiFi vs. Potenciómetros:** El ESP32 comparte los recursos del convertidor analógico (ADC2) con el módulo WiFi.
 - *Limitación:* Si activas el WiFi, **los potenciómetros dejarán de leer correctamente**. Para proyectos de síntesis pura, mantén el WiFi apagado.
 -  **Truco para Cargar Código (Upload Mode):** Si tienes problemas al subir un sketch, gira el **Potenciometro 3 al MÁXIMO**. Esto libera el pin 12 (que es un pin de arranque crítico) para permitir la programación.
 -  **Modificación de Hardware (Diodos):** En el diseño original había diodos que afectaban el voltaje; en tu placa v2 estos han sido reemplazados por "puentes" (jumpers) para garantizar el funcionamiento óptimo.
-

3. Fabricación y Hardware Libre

Este es un proyecto Open Source. En el repositorio encontrarás:

- **Archivos Gerber:** Puedes mandar a fabricar tus propias placas en servicios como JLCPCB o PCBWay.
- **Esquemáticos:** Siéntete libre de corregir los errores mencionados y crear la v2.5 o v3.0. ¡La evolución depende de la comunidad!