

Digital Synth PRA32-U

by ISGK Instruments <https://risgk.github.io/>



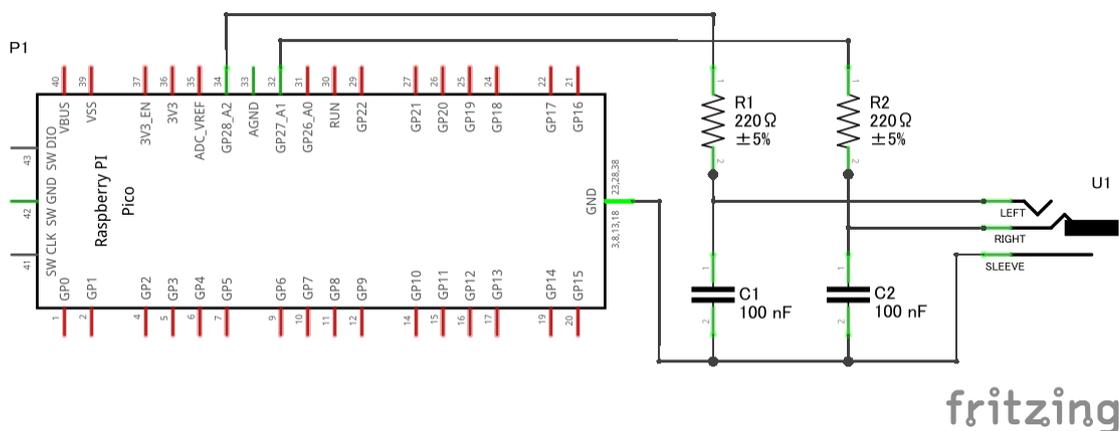
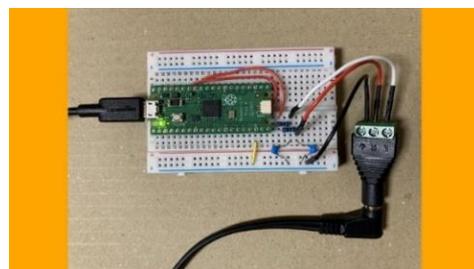
Raspberry Pi Pico を使った、誰でも自由に作れるシンセサイザー（USB MIDI 音源）です。抵抗、コンデンサ、オーディオ・ジャックがあれば「PWM オーディオ版」を、市販のオーディオ用 DAC 基板（「Pico Audio Pack」など）を使えば高音質な「DAC 版」を作れます。部品を追加することで、単体で音色編集や演奏が可能な「PRA32-U with Panel」も作れます。

主な仕様

- 制御方法：USB MIDI（または UART、MIDI）
※PC アプリ「PRA32-U CTRL」で 46 個のパラメーターを調整可能
- オーディオ出力：PWM オーディオ（または I2S）、48 kHz/16 bit/ステレオ
- 開発環境：Arduino IDE（Arduino-Pico） ●ライセンス：CC0（フリー）

PWM オーディオ版

抵抗、コンデンサ、オーディオ・ジャックが必要。
USB 接続の仕方によってはノイズが目立つので、
UART、MIDI 制御を推奨。



DAC 版

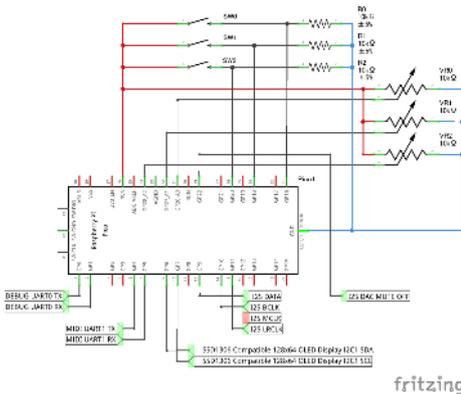
高音質、フラッシュに音色保存が可能。市販の I2S DAC 基板が必要（Pimoroni の「Pico Audio Pack」を推奨、「GY-PCM5102」モジュールも使用可）。



PRA32-U with Panel

現在プロトタイプで、今後仕様変更の可能性あり。

DAC 版にタクト・スイッチ、可変抵抗器、SSD1306 互換ドライバ対応 128x64 OLED ディスプレイを追



加。写真の製作例では、Seed Studio 「Grove Shield for Pi Pico（+足長 2x20 ピンソケット 2 個）」 「Buttons」 「Rotary Angle Sensors」 「OLED Display 0.96 inch」と necobit 電子「MIDI Unit for GROVE」を使用。ADC が USB MIDI 通信ノイズの影響を受けるのを防ぐため、UART、MIDI 制御を推奨。

設計ノート

- デュアルコアの使用**：v2 では、ポリフォニックモードとパラフォニックモードで信号処理にコア 0 と 1 を使用。ラズパイ Pico の仕組み上、同じコードを並列処理する場合はメモリ共有による性能低下は小さいと判断。安定動作のため、発音遅延時間（レイテンシ）は 5.3 ms に増加した。
- 関数属性 `__attribute__((always_inline))`**：高速化のため、ほとんどの関数を強制的にインライン展開。
- マクロ `__not_in_flash_func()`**：信号処理コードをフラッシュメモリでなく、高速アクセス可能な SRAM に配置。エントリ関数だけにこのマクロを付ければ、インライン展開との合わせ技で、ほとんどの関数が SRAM に載る。TinyUSB や MIDI Library のコードはフラッシュに置かれてしまうが、XIP キャッシュに載ることを期待。
- 参考文献**：『ラズベリー・パイ Pico/Pico W 攻略本』CQ 出版社（2023 年）