The PJRC bootloader chip that Teensy boards use is proprietary and not publicly available, so you can't just drop it into a custom board.

Alternatives:

- NXP’s native USB bootloader
    - The i.MX RT1062 has a built-in **ROM bootloader**

    You can manually enter bootloader mode by:

    1. Setting `BOOT_MODE0 = 1` (e.g., pull-up)
    2. Asserting `BOOT_MODE1 = 0` (e.g., pull-down)
    3. Holding down a reset or boot button sequence. Then, flash firmware over USB using NXP's MCU Boot Utility or serial Downloader.

    Downside: Not user-friendly for consumers unless you add a GUI.
- Add a small helper MCU for USB boot\
    - Add an **ATmega32U4**, **STM32F0**, or similar inexpensive MCU.
    - This helper can:
        - Emulate the PJRC-style bootloader behavior (enter boot mode, reset control, USB serial).
        - Optionally, implement DFU or a custom bootloader protocol.
    - Requires firmware development for this helper MCU.

    Best if: You want plug-and-play USB updating without depending on NXP tooling.
- Use the Teensy 4.1 module directly
    - For early prototypes, you can use the entire **Teensy 4.1 module** and plug it into your custom board via castellated headers or sockets.
    - PJRC handles the bootloader; you design only the peripherals around it.

    Best for: MVPs and testing before going full custom.

Recommended:

Since we're working on a startup audio interface:

- For MVP or early batch: **Use Teensy 4.1** modules with custom carrier boards.
- **For production: Use NXP’s bootloader with a clean UX, or add a helper MCU to manage boot/firmware loading via USB.**