

Projeto de Controlador MIDI Baseado em Raspberry Pi Pico

Aluno: Marcelo de Araújo

Email: marcelo.araujo02@aluno.ifce.edu.br

Introdução

A evolução das interfaces musicais digitais (MIDI) trouxe novas possibilidades para a criação de instrumentos personalizados e controladores musicais. Este projeto visa o desenvolvimento de um controlador MIDI utilizando o micro controlador Raspberry Pi Pico. A solução proposta combina simplicidade, custo acessível e capacidade de expansão, permitindo a interação com diversos dispositivos MIDI e softwares musicais.

Objetivos do Projeto

1. Desenvolver um Controlador MIDI Funcional Utilizando o Raspberry Pi Pico

O principal objetivo deste projeto é criar um controlador MIDI autônomo e funcional baseado no micro controlador Raspberry Pi Pico. Esse dispositivo servirá como uma interface prática para músicos, produtores e entusiastas de áudio, permitindo a interação direta com sintetizadores, computadores ou outros dispositivos compatíveis com MIDI. A escolha do Raspberry Pi Pico como plataforma central deve-se à sua acessibilidade, flexibilidade de programação e suporte a interfaces de hardware como GPIO, UART e I2C.

2. Implementar um Teclado MIDI com 13 Teclas, Controle de Oitava e Funções Avançadas

O controlador será equipado com um teclado físico composto por 13 teclas, correspondendo a uma oitava cromática completa, incluindo notas naturais e acidentes (sustenidos e bemóis). Cada tecla será configurada para enviar comandos MIDI, permitindo ao usuário tocar notas individuais e criar melodias. Além disso, o sistema oferecerá controle dinâmico de oitava, possibilitando ao usuário deslocar o registro das notas para agudos ou graves com facilidade, expandindo a usabilidade do teclado.

3. Garantir Comunicação MIDI Estável e Flexível via UART e I2C

A comunicação MIDI será implementada utilizando a interface UART do Raspberry Pi Pico, garantindo compatibilidade com dispositivos MIDI convencionais. Adicionalmente, será incorporado suporte à interface I2C, permitindo futuras expansões e integração com dispositivos periféricos, como controladores externos, displays ou módulos adicionais. Essa abordagem modular visa assegurar a escalabilidade e a adaptabilidade do projeto às demandas dos usuários.

4. Integrar Feedback Visual ao Usuário Utilizando LEDs Indicadores

O controlador contará com um sistema de feedback visual baseado em LEDs para informar o usuário sobre o estado atual do dispositivo e interações realizadas. Por exemplo, LEDs serão usados para indicar a troca de oitavas, pressionamento de teclas, alterações de parâmetros e outros eventos relevantes. Essa funcionalidade melhora a experiência do usuário ao fornecer respostas visuais claras e imediatas às ações realizadas.

Hardware Necessário para Controlador MIDI com Raspberry Pi Pico

Micro controlador

O Raspberry Pi Pico será o núcleo central do seu controlador MIDI. Ele é compatível com o SDK da Raspberry Pi, o que permite a programação através de C/C++ ou MicroPython. O Raspberry Pi Pico tem a capacidade de lidar com múltiplos pinos GPIO, comunicação UART para a interface MIDI, e I2C para integração de dispositivos externos.

Entradas do Teclado

O teclado será composto por 13 botões (ou teclas) que representarão as notas musicais do padrão MIDI. Cada botão será conectado a um pino GPIO específico no Raspberry Pi Pico. Estes botões também devem ser configurados com resistores de **pull-up** ou **pull-down** para garantir a leitura correta do estado dos botões.

- **GPIO 15:** NOTE_C_LOWER (Dó baixo)
- **GPIO 14:** NOTE_D_FLAT (Ré bemol)
- **GPIO 13:** NOTE_D (Ré)
- **GPIO 12:** NOTE_E_FLAT (Mi bemol)
- **GPIO 11:** NOTE_E (Mi)
- **GPIO 10:** NOTE_F (Fá)
- **GPIO 16:** NOTE_G_FLAT (Sol bemol)
- **GPIO 17:** NOTE_G (Sol)
- **GPIO 18:** NOTE_A_FLAT (Lá bemol)
- **GPIO 19:** NOTE_A (Lá)
- **GPIO 20:** NOTE_B_FLAT (Si bemol)
- **GPIO 21:** NOTE_B (Si)
- **GPIO 22:** NOTE_C_UPPER (Dó superior)

Controle Adicional

- **Botão Encoder (GPIO 9):** Um botão adicional para alternar entre diferentes funções, como mudança de oitava. Esse botão permite ao usuário manipular a oitava das notas tocadas.
- **Rotary Encoder (opcional):** Pode ser integrado para ajuste fino de parâmetros MIDI, como volume ou modulação. O rotary encoder é um dispositivo que permite ajustes contínuos, ideal para o controle de parâmetros de áudio e MIDI.

Saída MIDI

- **Interface UART MIDI:** Utilizando os pinos UART do Raspberry Pi Pico (TX e RX), você pode configurar uma interface MIDI que envia e recebe mensagens MIDI. O conector MIDI pode ser um **DIN de 5 pinos** tradicional ou um **adaptador USB-MIDI**.
 - **TX (GPIO 0):** Transmitir dados MIDI.
 - **RX (GPIO 1):** Receber dados MIDI.

Feedback Visual

- **LED Indicador (GPIO 25):** Um LED será utilizado para fornecer feedback visual ao usuário. Este LED pode ser usado para indicar a recepção de mensagens MIDI ou para representar o status do controlador (por exemplo, quando uma nota é pressionada ou quando uma mensagem MIDI é recebida).

Comunicação I2C

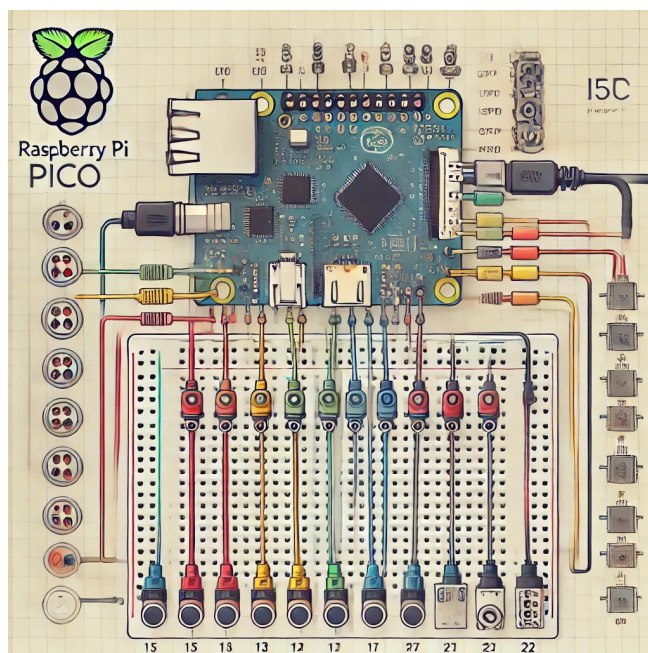
- **Barramento I2C:** O Raspberry Pi Pico tem suporte à comunicação I2C, o que possibilita a integração com dispositivos externos. Os pinos GPIO 2 (SDA) e GPIO 3 (SCL) serão configurados para esta comunicação, permitindo a conexão com sensores ou displays.
 - **GPIO 2 (SDA):** Dados para a comunicação I2C.
 - **GPIO 3 (SCL):** Clock para a comunicação I2C.

Alimentação

- **Fonte de 5V ou USB:** O Raspberry Pi Pico pode ser alimentado de duas formas: via **cabo USB** ou com uma **fonte externa de 5V**. O cabo USB é a forma mais prática para alimentação e programação do dispositivo.

Extras Opcionais

- **Display I2C:** Para feedback adicional ao usuário, como exibição de informações sobre as notas ou parâmetros MIDI. Embora não esteja implementado no código atual, a adição de um display I2C pode ser útil para projetos futuros.
- **Amplificador ou DAC:** Caso o projeto seja expandido para incluir uma saída de áudio analógica, você pode integrar um amplificador ou DAC (Conversor Digital para Analógico) para gerar sinais de áudio.



Funcionalidades Implementadas

1. Detecção de Teclas

Cada tecla do controlador MIDI é associada a um pino GPIO específico no Raspberry Pi Pico. Quando uma tecla é pressionada, o micro controlador detecta o evento de pressionamento e envia uma mensagem MIDI correspondente ao evento. Essas mensagens podem ser de dois tipos principais:

- **"NOTE_ON"**: Envia-se quando uma tecla é pressionada, indicando que uma nota foi acionada.
- **"NOTE_OFF"**: Envia-se quando a tecla é solta, sinalizando que a nota foi desativada.

Essa implementação permite uma interação básica com o instrumento musical, onde cada tecla corresponde a uma nota musical específica.

2. Controle de Oitava

Um botão Encoder foi integrado ao sistema para permitir ao usuário alterar a oitava de reprodução das notas MIDI. Isso possibilita o ajuste dinâmico da altura das notas tocadas, indo para oitavas mais graves ou agudas, dependendo da necessidade do usuário. A manipulação desse controle proporciona flexibilidade na performance, adaptando-se às preferências do músico.

3. Arbitragem de Canal

Foi implementada uma lógica de arbitragem de canal para gerenciar até quatro vozes simultâneas. O sistema é capaz de enviar até quatro mensagens de "NOTE_ON" diferentes para diferentes canais MIDI, permitindo que múltiplas notas sejam tocadas ao mesmo tempo sem conflito. A arbitragem garante que cada canal de voz receba e interprete as mensagens MIDI de forma eficiente e sem interferência entre si.

4. LED Indicador

Um LED indicador foi adicionado ao controlador para fornecer feedback visual ao usuário. Esse LED serve para indicar vários estados do sistema, como:

- **Alteração de Função:** Quando o usuário altera funções, como a seleção de uma oitava ou a ativação de um modo específico, o LED pisca ou acende para sinalizar a mudança.
- **Recepção de Mensagens MIDI:** O LED também pode piscar quando o controlador recebe uma mensagem MIDI, ajudando a confirmar a comunicação entre dispositivos. Esse feedback visual melhora a experiência do usuário, tornando a operação mais intuitiva e transparente.

Conclusão

Este projeto ilustra como o Raspberry Pi Pico pode ser aproveitado para desenvolver um controlador MIDI altamente eficiente e funcional, ideal para uso em ambientes musicais. O design adotado é modular, oferecendo flexibilidade para futuras expansões e personalizações. Isso significa que, no futuro, poderão ser adicionados novos componentes e funcionalidades ao sistema, como displays gráficos para visualização de informações em tempo real, sensores para detectar gestos ou pressão adicional nas teclas, ou até mesmo saídas de áudio para gerar som diretamente do controlador.

A simplicidade do hardware utilizado é um dos grandes atrativos, já que o Raspberry Pi Pico é um micro controlador de baixo custo, mas de alta performance, permitindo a construção de um controlador MIDI sem a necessidade de componentes complexos ou caros. A modularidade e o baixo custo tornam a solução acessível a uma ampla gama de entusiastas, desde músicos até engenheiros e makers que buscam explorar a interação entre eletrônica e música.

Além disso, a flexibilidade do código-fonte é um ponto forte do projeto, permitindo que o firmware seja facilmente modificado para se adequar às necessidades de cada usuário. Isso possibilita personalizações na forma como o controlador MIDI responde a entradas, como teclas pressionadas, ou como ele envia e recebe mensagens MIDI, facilitando a adaptação para diferentes estilos e preferências musicais.

O próximo passo no desenvolvimento do projeto incluirá melhorias no firmware, com o objetivo de otimizar a performance, adicionar novos recursos e corrigir eventuais falhas. Também será explorada a integração com interfaces gráficas, que poderão facilitar a configuração do dispositivo e a visualização de parâmetros, além de possibilitar um controle mais intuitivo sobre as funções do controlador. Por fim, serão realizados testes em aplicações reais de produção musical para validar o desempenho do controlador em situações práticas e identificar áreas para mais aprimoramentos.