



2 Entrega BigDogLag

PROJETO INSTRUMENTO MUSICAL

CURSO: Sistemas Embarcados

Professores: Rafael Lima, Alexandre Sales e Moacy Pereira

Alunos: Emily Xavier Gurjão e Nicholas Gomes da Costa

Projeto: Instrumento Musical

Campina Grande, PB

27/03/2025

1) Planejamento e Definição do Tema

PROJETO INSTRUMENTO MUSICAL

1.1 Contextualização e Justificativa do Tema

Com os avanços da tecnologia, novos instrumentos digitais têm surgido, permitindo maior interatividade e personalização na criação musical. Este projeto busca unir tecnologia e criatividade ao desenvolver um instrumento musical digital interativo, baseado no Raspberry Pi Pico.

O sistema utiliza um joystick analógico e botões físicos para permitir ao usuário controlar e modificar melodias em tempo real. Diferente de instrumentos convencionais, aqui o usuário pode experimentar variações dinâmicas de frequência, permitindo uma abordagem inovadora para a composição e performance musical.

Como funciona?

O projeto se baseia no uso de um buzzer controlado por PWM, gerando notas musicais que podem ser alteradas pelo joystick. O joystick modifica a frequência das notas, permitindo efeitos sonoros únicos, enquanto os botões possibilitam alternar entre diferentes melodias e controlar a reprodução. O sistema é programado em C, utilizando bibliotecas específicas para a leitura do joystick, acionamento dos botões e controle do som.

Por que este é um bom projeto?

- **Criatividade e Interatividade:** Permite aos usuários criarem variações musicais em tempo real, tornando a experiência imersiva e intuitiva.
- **Baixo Custo e Acessibilidade:** O Raspberry Pi Pico e os componentes utilizados são acessíveis e de fácil obtenção.
- **Flexibilidade e Expansão:** O código modular facilita a adição de novos recursos, como mais efeitos sonoros ou suporte a outros sensores.

Este projeto não apenas combina hardware e software de forma inovadora, mas também abre portas para experimentações artísticas, sendo uma ferramenta poderosa para aprendizado e exploração musical.

1.2 Objetivos do Projeto

1.2.1. Objetivo Geral

Desenvolver um instrumento musical digital interativo que permita ao usuário modificar sons em tempo real por meio de um joystick e botões físicos, proporcionando uma experiência inovadora na criação e execução musical.

1.2.2 Objetivos Específicos

- Criar um sistema musical interativo.
- Implementar controle de reprodução com botões.
- Permitir remixagem ao vivo alterando a frequência das notas.
- Aplicar conceitos de programação embarcada, incluindo ADC, PWM e interrupções.
- Desenvolver uma interface intuitiva para facilitar a interação do usuário com o instrumento.
- Explorar possibilidades de expansão do projeto, como integração com outros sensores ou armazenamento de melodias.

2) Desenvolvimento

Componentes Utilizados e Suas Funções

- **Raspberry Pi Pico:** Microcontrolador utilizado para gerenciar todas as interações do sistema.
- **Joystick Analógico:** Permite modificar a frequência das notas musicais em tempo real.
- **Botões físicos:** Controlam a troca de música e a reprodução (play/pause).
- **Buzzer Piezoelétrico:** Emite os sons musicais com variação de frequência controlada por PWM.
- **Resistores e fiação:** Utilizados para conectar os componentes ao Raspberry Pi Pico de maneira eficiente.

Bibliotecas Implementadas

O projeto foi estruturado em três bibliotecas principais que trabalham em conjunto para fornecer todas as funcionalidades do instrumento musical:

Joystick (joystickPi.h/.c)

Esta biblioteca é responsável por toda a interação com o joystick analógico. Seu funcionamento baseia-se em:

- Leitura contínua dos valores ADC dos eixos X e Y
- Filtragem dos valores para eliminar ruídos (usando média móvel)
- Normalização dos valores para faixas úteis (0-1 para volume, 0.5-1.5 para frequência)

CODIGO

Buzzer (buzzer.h/.c)

Controla a geração de sons através do buzzer piezoelétrico:

- Configuração automática do hardware PWM
- Geração precisa de tons com controle de frequência e volume
- Sistema de fila para sequências musicais

CODIGO

Botões (buttonPi.h/.c)

Gerencia todos os aspectos relacionados aos botões físicos:

- Implementa debounce por software (com temporização configurável)
- Oferece sistema de callbacks para eventos de pressionamento
- Suporta tanto polling quanto interrupções

CODIGO

Arquivo Principal (main.c)

A função principal (main.c) serve como o núcleo orquestrador do sistema, integrando perfeitamente todas as bibliotecas e gerenciando o fluxo completo da aplicação. Vamos detalhar profundamente sua estrutura e funcionamento:

```
[Inicialização]
|
|-[Configura Hardware]
|-[Carrega Músicas]
|-[Inicia Sistemas]
|
[Loop Principal]
|
|-[Leitura de Entradas]
| |-[Joystick]
| |-[Botões]
|
|-[Processamento Musical]
| |-[Ajuste de Parâmetros]
| |-[Gerenciamento de Notas]
| |-[Aplicação de Efeitos]
|
|-[Atualização de Interface]
| |-[Console Serial]
| |-[Indicadores Visuais]
```

2.1 Inicialização

1. Configura todos os periféricos hardware
2. Carrega as melodias da memória flash
3. Inicializa os sistemas de controle

CODIGO

2.2 Loop Principal

O sistema opera em um loop contínuo que:

- Leitura de Entradas: Verifica joystick e botões a cada 10ms
- Processamento Musical:
 - a. Calcula os efeitos do joystick nas notas
 - b. Gerencia o timing preciso das notas
 - c. Aplica efeitos em tempo real
- Atualização da Interface:
 - d. Exibe informações no console serial
 - e. Atualiza LEDs indicadores (se presentes)

CODIGO

3. Funcionamento Detalhado

3.1 Fluxo de Controle Musical

1. Seleção de Música: O botão A cicla através do array de melodias disponíveis
2. Controle de Reprodução: O botão B alterna entre play/pause
3. Modificação em Tempo Real: O joystick controla:
 - Eixo X: Fator de multiplicação de frequência (0.5x a 1.5x)
 - Eixo Y: Nível de volume (0% a 100%)

3.2 Processamento de Áudio

O sistema utiliza PWM para gerar formas de onda quadradas básicas. A frequência é calculada como:

$\text{freq_actual} = \text{freq_base} * \text{joystick_x_factor}$

O duty cycle do PWM é ajustado conforme a posição do eixo Y do joystick para controlar o volume percebido.

3.3 Gerenciamento de Melodias

As músicas são armazenadas como estruturas contendo:

CODIGO

Esta implementação da main demonstra uma arquitetura robusta e bem organizada, com:

- Separação clara de responsabilidades
- Baixo acoplamento entre módulos

- Alta coesão interna
- Facilidade de manutenção e expansão

O código completo, com todas as otimizações e tratamentos de erro detalhados, está disponível no repositório do projeto, incluindo versões comentadas linha por linha para facilitar o entendimento e a modificação por outros desenvolvedores.

3) Conclusão

O projeto atingiu plenamente seus objetivos principais, como reprodução de melodias com ajuste dinâmico de frequência e volume através do joystick, controle de play/pause e seleção de músicas via botões físicos.

Resultados Alcançados:

- Sistema musical funcional com controle intuitivo
- Bibliotecas reutilizáveis para joystick, botões e buzzer
- Interface serial para debug e feedback
- Código bem documentado e modular

A arquitetura modular, com bibliotecas dedicadas para cada componente (joystick, botões e buzzer), mostrou-se eficiente tanto para desenvolvimento quanto para manutenção do código.

A equipe que realizou o desenvolvimento dessas bibliotecas demonstrou organização e cuidado excepcionais no desenvolvimento deste projeto. Os códigos apresentam comentários claros que facilitam o entendimento da lógica por trás de cada biblioteca. Todas as funções foram implementadas de forma concisa, com objetivos bem definidos e específicos.

O trabalho revela um profissionalismo notável, principalmente na documentação completa desenvolvida para cada biblioteca. A dupla mostrou prudência ao elaborar manuais detalhados, garantindo que qualquer desenvolvedor possa compreender e utilizar facilmente os módulos criados.

A estrutura do projeto foi cuidadosamente planejada, com uma divisão lógica entre as diferentes funcionalidades. Cada componente foi encapsulado de forma coerente, permitindo manutenção e expansão simplificadas. A clareza dos comentários e a objetividade das implementações demonstram uma abordagem metodológica e profissional em todo o processo de desenvolvimento.

Diante dessas bibliotecas notou-se algumas melhorias podiam ser realizadas nas mesmas, como por exemplo: para a JoystickPi, poderíamos implementar rotinas de calibração automática para compensar variações nos componentes analógicos, além de adicionar suporte a gestos - como reconhecimento de movimentos rápidos que poderiam ativar efeitos especiais. A ButtonPi poderia evoluir para incluir tratamento de pressionamento longo e

combinações de botões, ampliando as possibilidades de interação. Já a BuzzerPi tem potencial para incorporar geração de acordes, efeitos sonoros mais elaborados e até mesmo síntese básica de diferentes formatos de onda.

Na parte do projeto, algumas melhorias foram idealizadas como por exemplo :

1. Interface Gráfica com Display OLED

- Adição de um display I2C para mostrar informações como:
 - Nome da música atual
 - Frequência e volume
 - Visualização do espectro de frequências

2. Efeitos Sonoros Avançados

- Implementação de efeitos como:
 - Vibrato (variação periódica de frequência)
 - Echo (repetição de notas com decaimento)
 - Portamento (transição suave entre notas)

3. Expansão do Repertório

- Desenvolvimento de um sistema para carregar melodias de arquivos
- Suporte a formatos como MIDI ou notação musical simplificada

A documentação técnica completa está disponível para consulta no repositório GitHub do projeto <https://github.com/nicholas7821/ProjetoFinal>. Esta base de conhecimento foi cuidadosamente organizada para todos que desejam utilizar esse código ou para aprimorar o mesmo .