

Projeto Final: Mini Piano com Joystick + Buzzer usando FreeRTOS

Leandro Martins Tosta (2232510) , Eiti Parruca Adama (2232472)

¹Graduação em Engenharia da Computação

Universidade Tecnológica Federal do Paraná (UTFPR) – Apucarana, PR – Brasil

Resumo. *Este projeto apresenta a implementação de um mini piano digital baseado em Arduino Mega, utilizando um joystick analógico para selecionar notas musicais e um buzzer para reprodução sonora. O sistema emprega o sistema operacional de tempo real FreeRTOS para gerenciamento multitarefa eficiente, com uso de filas e semáforos. Além da funcionalidade de execução de notas, o projeto permite a gravação e posterior reprodução de sequências musicais. O LCD 16x2 exibe informações sobre a nota atual, frequência e modo de operação. O projeto demonstra o uso de sistemas embarcados e técnicas de controle em tempo real com aplicações didáticas e interativas.*

1. Introdução

Sistemas embarcados estão cada vez mais presentes em nosso cotidiano, desde dispositivos médicos até equipamentos automotivos. A utilização de sistemas operacionais de tempo real, como o FreeRTOS, permite dividir tarefas de forma mais organizada e eficiente. Este projeto tem como objetivo desenvolver um sistema interativo e musical, um mini piano, que permita o usuário tocar notas, visualizar informações no LCD e gravar sequências musicais. A proposta busca reforçar conceitos de multitarefa, comunicação entre tarefas e uso de periféricos com Arduino.

2. Descrição do Problema

O desafio proposto consiste em criar um sistema embarcado musical interativo que:

- Permita a seleção e execução de 7 notas musicais (C a B).
- Use o joystick como controlador de entrada.
- Reproduza as notas com um buzzer.
- Exiba informações em tempo real no LCD.
- Permita gravação de uma sequência de notas e reprodução posterior.
- Utilize recursos do FreeRTOS: multitarefas, filas e semáforos.

O sistema deve ser responsivo, estável e intuitivo para o usuário, mesmo operando em tempo real.

3. Aplicações em Sistemas Reais

Sistemas embarcados com interface musical e controle por joystick, como o projeto Mini Piano apresentado neste trabalho, têm aplicabilidade em diversas áreas. Na educação, por exemplo, sistemas similares podem ser empregados em ambientes didáticos para ensino de fundamentos de eletrônica, programação e teoria musical de forma interativa e lúdica. Segundo Silva [Silva 2023], a adoção de sistemas embarcados em contextos educacionais

favorece o aprendizado prático e estimula o desenvolvimento de competências multidisciplinares.

Além disso, o uso de multitarefa com FreeRTOS, semáforos e filas, conforme implementado neste projeto, é prática comum em aplicações industriais reais. Em projetos mais avançados, esses recursos são aplicados em dispositivos como teclados musicais eletrônicos, sistemas de alerta sonoro personalizados e controladores de interface homem-máquina (IHM). O trabalho de Reis [Reis 2024], por exemplo, demonstra a importância de sistemas embarcados com FreeRTOS na manutenção preditiva e coleta de dados em tempo real, destacando a robustez dessa arquitetura.

Outro campo promissor é o de reabilitação motora. Sistemas interativos com som e resposta a movimentos, como o joystick, podem ser utilizados para terapias baseadas em feedback sensorial, conforme observado em iniciativas de ensino adaptativo e estimulação cognitiva. Esses tipos de aplicações reforçam como projetos aparentemente simples podem servir de base para soluções de alto impacto.

Adicionalmente, a documentação aberta de projetos similares, como o Laboratório 2 e 3 da disciplina de Sistemas Embarcados da UTFPR [iRocktys 2025, Tosta and Adama 2025a], demonstra o valor pedagógico desses experimentos no ensino superior de engenharia e tecnologia.

4. Implementação em hardware

Para a montagem em hardware, foram utilizados os seguintes materiais:

- Arduino Mega
- Módulo joystick analógico com botão integrado
- Display LCD 16x2
- Potenciômetro de $10k\Omega$ para ajuste de contraste do LCD
- Buzzer
- Resistor de 220Ω para o backlight do LCD
- jumpers para as conexões
- Protoboard para montagem dos componentes
- Biblioteca `Arduino_FreeRTOS.h`

As conexões foram realizadas de forma a evitar conflitos entre sinais: o LCD teve VSS em GND e VDD em 5 V, VO ao pino central do potenciômetro, RS em D7, RW em GND, E em D8 e D4–D7 em D9–D12 do Arduino, com o backlight alimentado pelo resistor de 220Ω , ligado a 5 V; o joystick alimentou-se em 5 V e GND, com VRx em A0, VRy em A1 e o botão SW em D2; o buzzer teve o terminal positivo em D3 e o terminal negativo em GND. Essa configuração é mostrada na Figura 1.

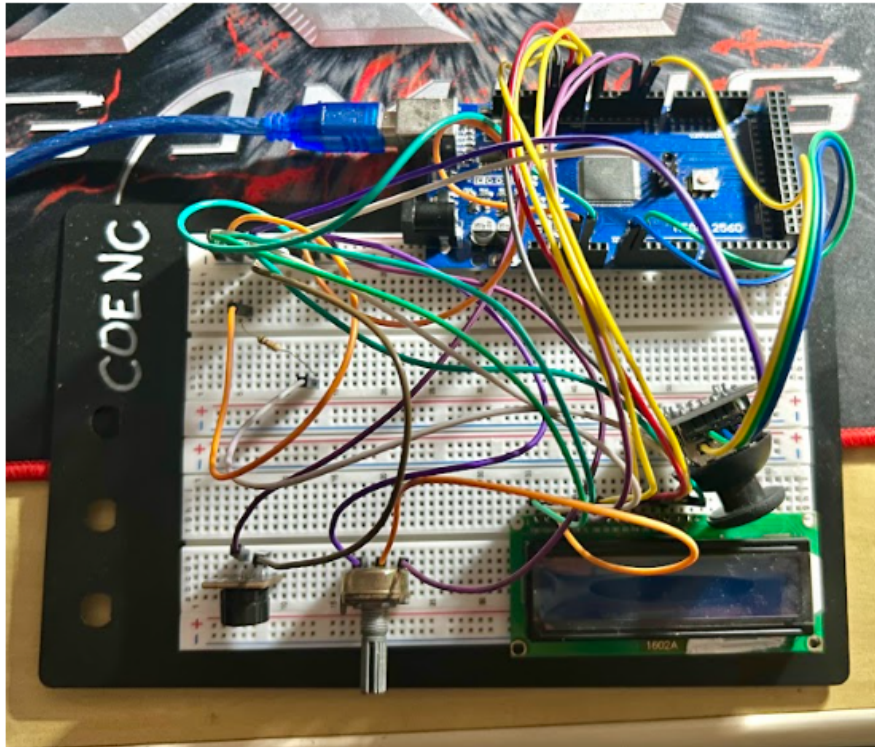


Figura 1. Implementação em hardware.

Fonte: Autoria própria (2025).

5. Implementação em software

5.1. Uso de FreeRTOS e Organização por Tarefas

O projeto foi estruturado com o uso do FreeRTOS para distribuir a lógica do sistema em tarefas cooperativas, que funcionam de forma concorrente. Ao invés de depender da função 'loop()', cada responsabilidade do sistema foi separada em uma **task** dedicada. Foram implementadas quatro tarefas principais, descritas a seguir:

1. Tarefa de Interface (taskInterface**)**

Esta tarefa é responsável por atualizar periodicamente o display LCD com informações relevantes ao usuário: nota musical selecionada, frequência da nota em Hertz e o modo atual de operação (NORMAL, GRAVANDO ou PLAY). Utiliza um semáforo (`lcdMutex`) para garantir exclusão mútua no acesso ao LCD, evitando conflitos com outras tarefas que também possam acessá-lo.

- Prioridade: 1
- Período: 250 ms
- Recurso compartilhado: LCD (com semáforo)

2. Tarefa do Joystick (**taskJoystick**)

Responsável por:

- Ler a posição analógica do joystick.
- Detectar mudanças de direção.
- Traduzir a direção em uma nota musical específica (índice e frequência).
- Enviar a nota para a fila de som (*filaSom*).
- Detectar cliques simples e duplos no botão:
 - 1 clique inicia/encerra gravação.
 - 2 cliques iniciam a reprodução da sequência gravada.
- Durante a gravação, a tarefa também envia a nota atual para a fila de gravação (*filaGravacao*).
- Prioridade: 2
- Período: 150 ms

3. Tarefa de Som (**taskSom**)

Monitora continuamente a fila de som (*filaSom*). Sempre que uma nova nota (frequência) é recebida, a tarefa aciona o buzzer com a função `tone()`, mantém o som por 500 ms e desliga com `noTone()`.

- Prioridade: 2
- Tipo: Bloqueante (espera por nova nota com `xQueueReceive`)

4. Tarefa de Reprodução (**taskReproducao**)

Ativada quando o estado do sistema muda para REPRODUZINDO. Ela lê sequencialmente todas as notas armazenadas na fila de gravação (*filaGravacao*), toca cada uma por meio da fila de som e reenvia a nota para a fila para manter a gravação persistente. Após tocar todas as notas, retorna automaticamente ao modo NORMAL.

- Prioridade: 1
- Período: 500 ms (verifica o estado)

5.2. Comunicação entre Tarefas

O sistema utiliza dois mecanismos principais de comunicação entre tarefas:

- **Semáforo binário:** `lcdMutex`, usado para sincronização no acesso ao display LCD, evitando escritas simultâneas.
- **Filas:**
 - *filaSom* (tipo `int`): transmite a frequência da nota a ser tocada para a tarefa de som.
 - *filaGravacao* (tipo `int`): armazena os índices das notas tocadas, possibilitando sua posterior reprodução.

Essa divisão de responsabilidades favorece modularidade, facilidade de manutenção e desempenho estável mesmo com múltiplos eventos ocorrendo quase simultaneamente.

5.3. Mapeamento das Notas

Cada direção do joystick foi mapeada para uma nota musical da escala diatônica (C a B), excluindo a direção central:

Direção do Joystick	Nota	Frequência (Hz)
CIMA	C	262
CIMA_DIREITA	D	294
DIREITA	E	330
BAIXO	F	349
BAIXO_ESQUERDA	G	392
ESQUERDA	A	440
CIMA_ESQUERDA	B	494

Tabela 1. Mapeamento das notas com o joystick

5.4. Gravação e Reprodução

- **1 clique** no botão do joystick: alterna entre Modo Normal e Gravação.
- **2 cliques** rápidos: entra no modo de Reprodução.
- A cada nota tocada durante a gravação, seu índice é enviado à fila `filaGravacao`.
- Ao iniciar uma nova gravação, a fila antiga é limpa com `xQueueReset(filaGravacao)`.

Para complementar a documentação deste projeto e facilitar a verificação dos resultados práticos, o código-fonte completo da implementação está publicamente disponível para consulta [Tosta and Adama 2025a]. Adicionalmente, foi produzido vídeo demonstrativo para ilustrar o funcionamento do protótipo em tempo real [Tosta and Adama 2025b].

6. Conclusão

O projeto Mini Piano com Joystick + Buzzer demonstrou de forma prática o uso de um sistema operacional em tempo real em um sistema embarcado lúdico e funcional. A estrutura modular com tarefas, filas e semáforos proporcionou um código organizado e eficiente. A interação via joystick permite uma experiência musical intuitiva, e a gravação/reprodução de notas agrega valor ao projeto. O uso de FreeRTOS proporcionou aprendizado importante sobre multitarefa, concorrência e comunicação entre tarefas, aspectos fundamentais no desenvolvimento de sistemas embarcados reais.

Referências

- iRocktys (2025). Repositório de códigos-fonte para a disciplina de sistemas embarcados. <https://github.com/iRocktys/Sistemas-embarcados/tree/main>. Acesso em: 03 jun. 2025.
- Reis, J. V. M. (2024). Desenvolvimento de um sistema embarcado inovador para manutenção preditiva e monitoramento em tempo real de vibração e temperatura. Technical report, Universidade Federal do ABC.

Silva, M. A. d. (2023). Desenvolvimento de um sistema embarcado para coleta de dados meteorológicos utilizando máquina de estados. Master's thesis.

Tosta, L. M. and Adama, E. P. (2025a). Código-fonte do projeto final: Mini piano com joystick + buzzer usando freertos. <https://github.com/iRocktys/Sistemas-embarcados/tree/main/Projeto-Final>. Acessado em: 7 de julho de 2025.

Tosta, L. M. and Adama, E. P. (2025b). Vídeo de demonstração do projeto final: Mini piano com joystick + buzzer usando freertos. https://drive.google.com/file/d/1_IjcnxcwU_l-pyCUBajQsjXOCKP3EbEg/view?usp=sharing. Vídeo online. Acessado em: 7 de julho de 2025.