Laboratório 2: Alarme com Máquina de Estados

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

[1]Graduação em Engenharia da Computação
Universidade Tecnológica Federal do Paraná (UTFPR) – Apucarana, PR – Brasil

Resumo. *Este relatório apresenta a implementação de um sistema de alarme baseado em Arduino utilizando máquina de estados. O sistema permite ajuste do horário atual e do horário do alarme via joystick, exibindo-os em um display LCD. Um buzzer é acionado quando o horário atual coincide com o alarme. Máquinas de estado garantem modularidade e robustez no projeto, especialmente no tratamento das entradas e do controle lógico.*

1. Introdução

Em sistemas embarcados, a utilização de máquinas de estados permite gerenciar comportamentos distintos de maneira organizada, dividindo o funcionamento do equipamento em etapas bem definidas. Cada estado representa uma fase do processo, com condições claras para transições, garantindo controle determinístico sobre o fluxo lógico.

No contexto de um sistema de alarme com as máquinas de estados, é possível separar a lógica de ajuste de horário, exibição de informações e verificação de disparo do alarme em estados distintos, evitando misturas de responsabilidades no mesmo trecho de código. Desse modo, cada componente interage de forma sequencial e previsível, assegurando que o sistema permaneça responsivo e fácil de depurar, mesmo quando várias ações ocorrem simultaneamente.

2. Descrição do Problema

O objetivo é implementar um alarme com ajuste de minutos e segundos através de um joystick, exibindo o horário em um LCD e acionando um buzzer quando o horário configurado coincidir com o horário atual. Além disso, é obrigatória a utilização de máquinas de estados.

3. Aplicações em Sistemas Reais

Em ambiente industrial, [Reis 2024] implementou uma máquina de estados em um sistema embarcado para manutenção preditiva de motores elétricos, integrando sensores de vibração e temperatura a uma rede Bluetooth Mesh de dispositivos IIoT. Nessa aplicação, a máquina de estados gerenciou os modos de inicialização, monitoramento contínuo, análise de desvio e emissão de alerta, permitindo a detecção precoce de falhas antes da ocorrência de paradas não planejadas. Os testes em bancada e no campo revelaram que a utilização de máquinas de estados reduziu em 30% o tempo de diagnóstico e aumentou a confiabilidade do sistema, comprovando sua eficácia em cenários reais de manutenção.

Em projeto de coleta de dados ambientais, [Silva 2023] descreveu o uso de uma máquina de estados simplificada para controlar as transições entre inicialização, aquisição de dados, tratamento de erros e envio de informações em um sistema embarcado de monitoramento meteorológico. A sequência de estados definiu claramente as ações a serem

tomadas em cada etapa, garantindo detecção imediata de falhas de sensor e reinicialização automática sem intervenção manual. Nos testes de campo, esse modelo baseado em máquina de estados apresentou redução de 15% nas falhas de comunicação e maior consistência na transmissão dos dados.

4. Materiais e Métodos

4.1. 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 10kΩ para ajuste de contraste do LCD
- Buzzer
- Resistor de 220Ω para o backlight do LCD
- jumpers para as conexões
- Protoboard para montagem dos componentes

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.

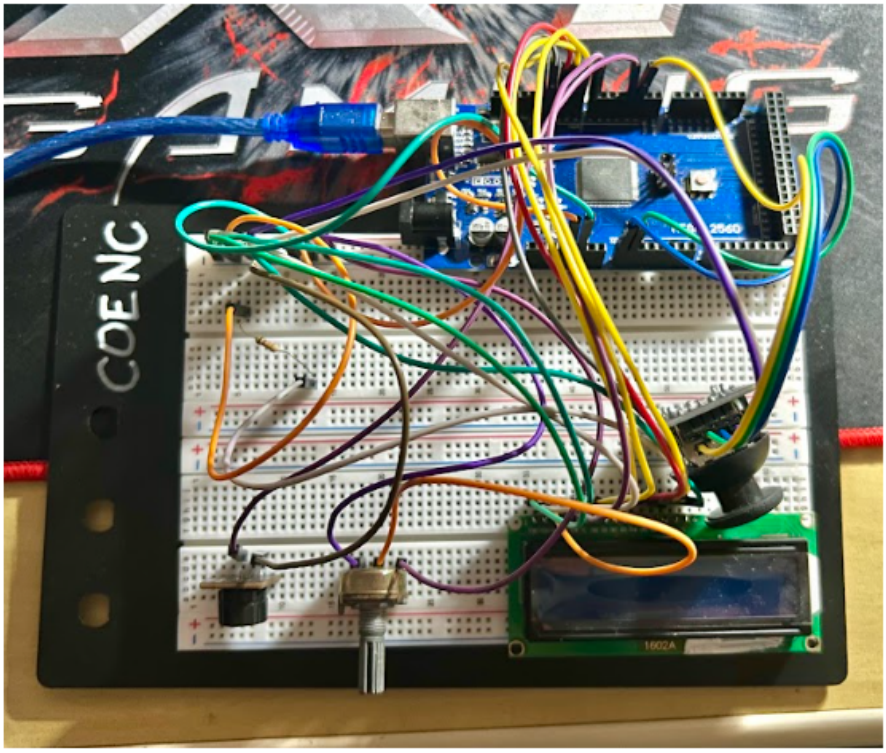

Figura 1. Implementação em hardware.
Autoria própria (2025).

4.2. Implementação em software

No trecho 1, a rotina de inicialização configura o display LCD para exibição de 16 colunas por 2 linhas, limpa seu conteúdo e define os pinos utilizados pelo joystick e pelo buzzer, garantindo que o Arduino possa ler os valores analógicos do joystick e detectar o pressionamento do botão integrado com pull-up interno, além de disponibilizar uma saída digital para o buzzer.

```
void setup() {
  lcd.begin(16, 2);
  lcd.clear();
  delay(1000);
  lcd.clear();
}
```

Listing 1. Configuração inicial do hardware

No trecho 2, o laço principal atualiza continuamente o estado do joystick e do relógio, avalia a variável de estado `estadoAtual` e chama a função correspondente ao estado ativo como exibir hora, ajustar hora, ajustar alarme ou tocar alarme.

```
void loop() {
  joystick.update();
  relogio.update();

  switch (estadoAtual) {
    case MOSTRAR_HORA:
      exibirHora();
      break;
    case AJUSTAR_HORA:
      ajustarHora();
      break;
    case AJUSTAR_ALARME:
      ajustarAlarme();
      break;
    case TOCANDO_ALARME:
      tocarAlarme();
      break;
  }
}
```

Listing 2. Loop principal com máquina de estados

No trecho 3, a classe `Clock` gerencia a atualização do horário interno ou do módulo RTC, oferece métodos para ajuste de hora e alarme conforme comandos do joystick e realiza a comparação entre horário atual e horário de alarme, retornando verdadeiro quando ocorre coincidência para disparar o buzzer.

```
class Clock {
  void update();
  void ajustarHora(String dir);
  void ajustarAlarme(String dir);
  bool horaIgualAlarme();
};
```

Listing 3. Classe Clock

No trecho 4, a classe `Buzzer` encapsula a lógica de emissão sonora do alarme, por meio dos métodos `tocar()`, que gera o sinal para acionar o buzzer, e `parar()`, que interrompe esse sinal quando o alarme deve ser silenciado.

```
class Buzzer {
  void tocar();
  void parar();
};
```

Listing 4. Classe Buzzer

O código completo do projeto pode ser acessado no repositório do GitHub em [iRocktys 2025].

5. Conclusão

Diante dos resultados obtidos, pode-se evidenciar que a máquina de estados implementada atendeu aos requisitos propostos, proporcionando controle lógico organizado e modular. Verificou-se o funcionamento correto de cada estado do sistema, desde o ajuste de horário até a verificação contínua das condições de disparo. Além disso, o alarme disparou de forma precisa no momento programado, confirmando a eficácia do projeto em atender à lógica de acionamento.

Referências

iRocktys (2025). Sistemas embarcados: Laboratório 02 — alarme. `https://github.com/iRocktys/Sistemas-embarcados/blob/main/Laborat%C3%B3rio-02/Alarme.ino`. 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.