

Professor: João Cláudio Elsen Barcellos, joao.barcellos@ifsc.edu.br

Estudante:

Exercício 1: Controlador para produção de cerveja artesanal

Objetivo e visão geral: Neste projeto propõe-se o desenvolvimento de um sistema embarcado voltado para o auxílio no processo de produção de cerveja. O sistema deve monitorar e controlar, em tempo real, a temperatura do mosto, acionando, quando necessário, um mixer, para homogeneizar a sua temperatura. Para isso, devem ser empregadas técnicas de controle como o *proportional-integral-derivative* (PID) ou ON/OFF com histerese. E o controle deve ser baseado em curvas — temperatura pelo tempo —, que podem ser definidas pelo usuário ou pré-configuradas no sistema.

A lógica de controle pode ser implementada a partir de uma máquina de estados gerada automaticamente, por meio de ferramentas como o Yakindu (atualmente chamada itemis CREATE), ou construída “do zero”. Nos casos em que o Yakindu for utilizado, espera-se que os *callbacks* sejam corretamente implementados no código-fonte. Do contrário, a máquina de estados (e todo o código-fonte) deverá ser estruturado com orientação a objetos, em C++, com destaque para o uso de métodos virtuais e polimorfismo. Nessa abordagem, deve-se ter uma versão embarcada (e.g., a ser usado num ESP32, Raspberry Pi Pico e similares) e outra versão para o PC, com troca de implementações via herança ou composição polimórfica. O sistema, como um todo, é apresentado de forma sucinta na Figure 1.

Requisitos mínimos:

1. Plataforma de controle:

- Microcontrolador de 32 bits com suporte a I²C, SPI e UART;
- Capacidade de operação em tempo real (bare-metal ou com RTOS).

2. Sensores e atuações:

- Leitura de temperatura via I²C;
- Controle de resistências (heaters) via *general purpose input/output* (GPIO) ou *pulse width modulation* (PWM);
- Controle de mixer via GPIO ou PWM.

3. Controle:

- Implementação de controle PID e/ou ON/OFF com histerese;
- Mixer deve ser ativado quando a diferença entre dois sensores de temperatura exceder 1 °C.

4. Máquina de estados:

- A máquina de estados pode ser gerada com o Yajin/itensis CREATE, com implementação dos *callbacks*;
- Alternativamente, o controle pode ser implementado manualmente em C++, desde que use métodos virtuais, polimorfismo e permita a sua simulação no PC;
- A versão simulada deve permitir que a lógica de decisão seja testada sem o hardware, via entradas simuladas e saídas controladas por abstrações.

5. **Curvas programáveis:**

- O sistema deve possuir uma curva padrão;
- Novas curvas podem ser inseridas via comunicação serial;
- As curvas devem descrever um perfil de temperatura em função do tempo.

6. **Interfaces:**

- Interfaces lógicas: GPIO, PWM, I²C, UART;
- Comunicação com o PC via UART.

7. **Detecção de erros:**

- O sistema deve identificar falhas (e.g, erro ao tentar configurar um módulo, erro ao ler um sensor);
- Deve alertar o usuário e desativar os atuadores para segurança.

8. **Documentação e repositório:**

- Projeto deve estar versionado em repositório no GitHub;
- Recomenda-se uso de Doxygen para documentação do código.

9. **Funcionalidade adicional:**

- Deve-se propor e implementar uma funcionalidade extra, como: alarme sonoro, registro de logs, visualização gráfica, modo de calibração, entre outros.

Sugestão de entregas parciais:

Entrega 1: Levantamento de requisitos, definição das curvas, modelagem da máquina de estados (manual ou Yajin) e diagramas de sistema.

Entrega 2: Implementação dos drivers de sensores e atuadores, controle básico (PID ou ON/OFF), e comunicação com o PC.

Entrega 3: Integração final com mixer funcional, detecção de falhas, funcionalidade adicional, documentação completa e demonstração.

Critérios de avaliação:

- Funcionamento dos módulos implementados;
- Clareza e organização do código;

- Uso de boas práticas de programação embarcada;
- Qualidade da documentação;
- Utilização efetiva do GitHub para controle de versão.

Observações:

- A implementação pode ser feita em C ou C++ (ou ambos);
- O sistema deve ser portátil e preparado para execução em uma plataforma embarcada real (e.g., ESP32 ou equivalente);
- É permitida a simulação parcial de dispositivos, desde que devidamente descrita e justificada;
- A lógica de controle deve ser validável tanto em ambiente embarcado quanto em simulação no PC.

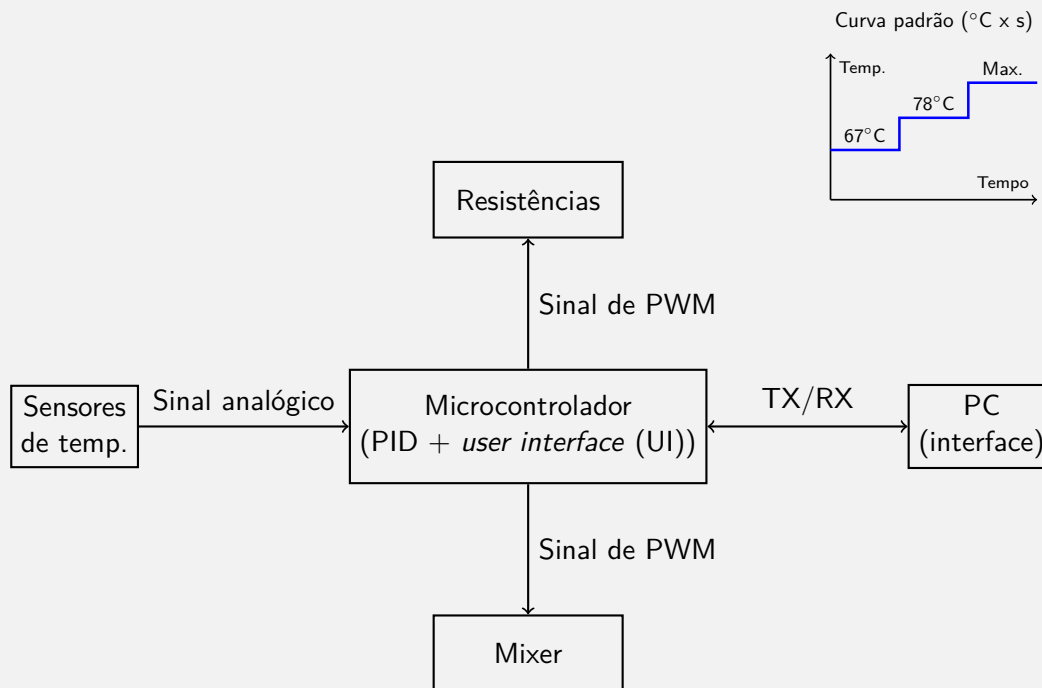


Figura 1: Diagrama geral do sistema de controle com curva padrão de temperatura.