

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

Estudante:

Exercício 1: Controlador para produção de cerveja artesanal

Objetivo e visão geral: Este projeto propõe o desenvolvimento de um sistema embarcado para auxiliar no processo de produção de cerveja artesanal. O sistema deve controlar, em tempo real, a temperatura do processo e acionar a agitação do mosto por meio de um mixer, conforme condições específicas. Para isso, devem ser empregadas técnicas de controle como PID ou ON/OFF com histerese. As decisões de controle devem se basear em curvas programáveis (temperatura versus 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 com a ferramenta Yakindu (atualmente chamada itemis CREATE), ou construída manualmente pelo aluno. Nos casos em que Yakindu for utilizado, espera-se que os métodos da interface gerada sejam corretamente implementados no código-fonte. Caso o aluno opte por construir a máquina de estados do zero, o sistema deverá ser estruturado usando conceitos de orientação a objetos em C++, com destaque para o uso de métodos virtuais e polimorfismo. Nessa abordagem, o sistema deve permitir uma versão embarcada (ESP32) e outra versão simulável no 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 GPIO ou 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 Yakindu/itemis CREATE, com implementação dos métodos obrigatórios da interface.

- Alternativamente, o controle pode ser implementado manualmente em C++, desde que use métodos virtuais, polimorfismo e permita a 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. **Deteção de erros:**

- O sistema deve identificar falhas como ausência de leitura dos sensores.
- Deve alertar o usuário e desativar os atuadores para segurança.

8. **Documentação e repositório:**

- Projeto deve estar versionado em repositório GitHub.
- Recomenda-se uso de Doxygen para documentação do código.

9. **Funcionalidade adicional:**

- O estudante deve 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 Yakindu) 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, deteção 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++ (preferencialmente, C++).
- O sistema deve ser portátil e preparado para execução em plataforma embarcada real (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.

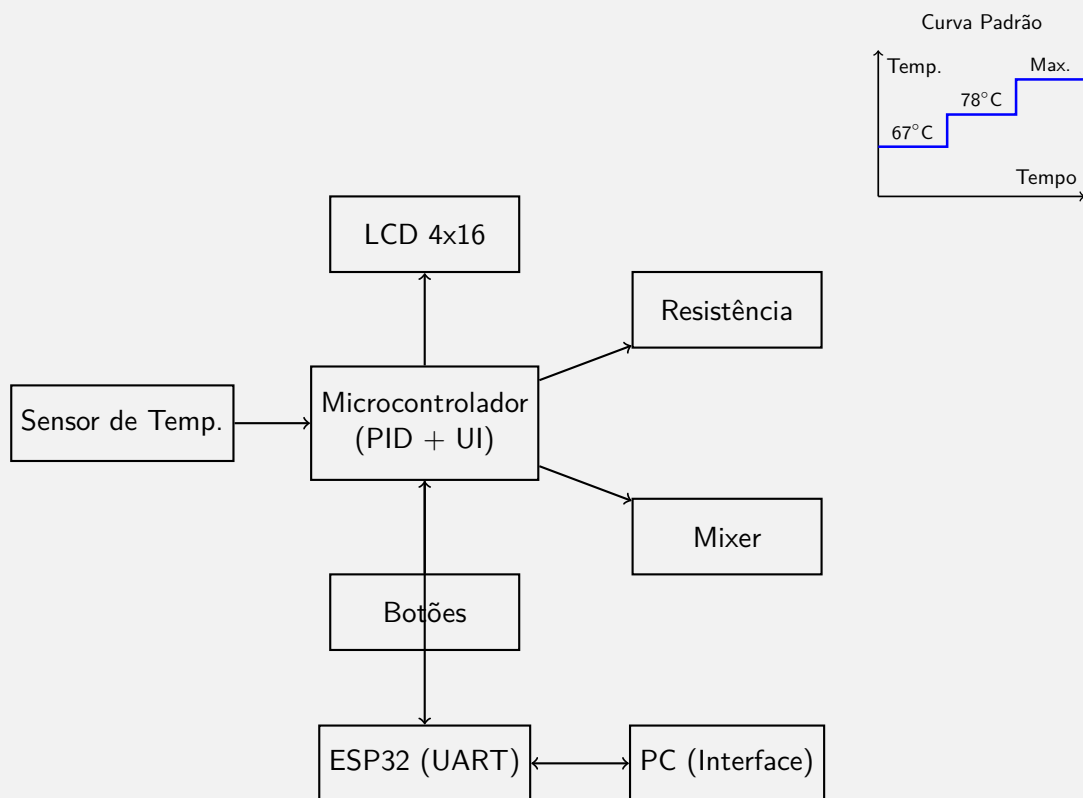


Figura 1: **FIXME:** Diagrama geral do sistema de controle com curva padrão de temperatura.