

Sistema Embarcado para Coleta e Análise de Dados do Solo com Lógica de Irrigação Otimizada

João Pedro Correa Bezerra
Esther Ewellin Campos Dias
Universidade Federal do Rio de Janeiro
joao.bezerra.20241@poli.ufrj.br
esther.ewellin.20241@poli.ufrj.br

8 de novembro de 2025

Resumo

Este artigo apresenta o projeto e a implementação de um protótipo de sistema embarcado para o monitoramento de parâmetros do solo. O sistema coleta dados de temperatura, umidade e salinidade a cada 15 minutos. Os dados são armazenados em uma lista encadeada e processados em ciclos de quatro horas: o maior e o menor valor de temperatura são descartados, e uma média dos valores restantes é calculada. Com base nessa média, uma lógica de decisão embarcada determina a necessidade de irrigação. Este trabalho detalha a arquitetura do sistema, as estruturas de dados utilizadas e apresenta um estudo de caso simulado para validar a funcionalidade do protótipo.

1 Introdução

O gerenciamento hídrico na agricultura é um desafio global. A irrigação inadequada, seja por excesso ou escassez, leva ao desperdício de água, redução da produtividade das culturas e degradação do solo, como a salinização. Muitos sistemas de irrigação atuais operam em horários fixos, sem considerar as condições reais do solo em tempo real.

Resolver este problema é relevante para garantir a segurança alimentar de forma sustentável. Um sistema que otimiza o uso da água baseado em dados reais do solo pode aumentar drasticamente a eficiência hídrica, economizar custos para o agricultor e minimizar o impacto ambiental.

Buscamos resolver este problema através do desenvolvimento de um sistema embarcado de baixo custo que realiza a coleta e análise de dados diretamente no campo. A proposta central é um ciclo de processamento de dados de quatro horas que filtra ruídos (removendo valores extremos) e calcula uma média estável para tomar uma decisão de irrigação mais confiável e autônoma.

2 Revisão da Literatura

A literatura sobre "smart farming" e irrigação de precisão é vasta. Smith et al. [1] propuseram um sistema baseado em IoT (Internet das Coisas) que utiliza nós sensores sem fio para enviar dados brutos para a nuvem. Embora eficaz, a dependência constante da nuvem para processamento aumenta o consumo de energia e os custos de transmissão.

Em [2], Chen e Li exploram o uso de lógica fuzzy em microcontroladores para a tomada de decisão de irrigação. Eles focam na complexidade da decisão, mas não abordam o pré-processamento de dados de sensores para remover outliers, o que pode levar a decisões errôneas baseadas em leituras anômalas.

O trabalho de Gupta [3] discute diferentes estruturas de dados para gerenciamento de dados de séries temporais em dispositivos com restrição de memória. Eles comparam arrays circulares (ring buffers) com listas para armazenamento temporário, concluindo que arrays são geralmente mais eficientes para amostras de tamanho fixo.

Nossa proposta se diferencia ao combinar o processamento de dados local (filtragem de outliers) com uma estrutura de dados específica (lista encadeada) e uma lógica de decisão embarcada,

visando um equilíbrio entre robustez dos dados e autonomia do dispositivo.

3 Proposta (Experimentação)

Nosso protótipo é um sistema embarcado (simulado em um ambiente Arduino C++) que coleta três parâmetros do solo: temperatura, umidade e salinidade. O sistema armazena 16 amostras (coletadas a cada 15 minutos, totalizando 4 horas) e, ao final do ciclo, processa esses dados para gerar uma média filtrada e tomar uma decisão de irrigação.

3.1 Estruturas de Dados

A requisito do projeto especificou o uso de uma lista encadeada. Utilizamos uma lista encadeada simples (*singly-linked list*) onde cada nó armazena um *struct* contendo os três valores (temp, umidade, salinidade). A cada 15 minutos, um novo nó é alocado dinamicamente (via **new**) e adicionado ao início da lista.

Embora o uso de alocação dinâmica em microcontroladores possa levar à fragmentação da memória, esta estrutura foi escolhida para atender aos requisitos. Uma alternativa mais robusta para um produto final seria um array estático de 16 posições, que ofereceria desempenho previsível e sem sobrecarga de alocação.

O processamento ao final das 4 horas (16 amostras) envolve:

1. Iterar pela lista para encontrar os nós com a maior e menor temperatura.
2. Iterar novamente, somando os valores de todos os nós, exceto os dois identificados.
3. Calcular a média dos 14 valores restantes.
4. Deletar todos os 16 nós (**delete**) para liberar a memória e evitar vazamentos.

3.2 Uso da Linguagem

O sistema foi implementado em C++ no framework do Arduino. O tempo é gerenciado de forma não-bloqueante usando a função `millis()`, permitindo que o microcontrolador execute outras tarefas (embora não implementadas neste protótipo). As estruturas **struct** foram usadas para organizar os dados. A lógica de decisão é uma

função simples que retorna uma **String** (representando o comando para o atuador de irrigação), que é então impressa na porta Serial.

3.3 Esquema Eletrônico

O sistema completo exigiria um microcontrolador (como um ESP32 ou Arduino Uno), um sensor de umidade do solo (ex: capacitivo), um sensor de temperatura (ex: DS18B20) e um sensor de salinidade (condutividade elétrica). A saída da lógica de decisão controlaria um relé, que por sua vez acionaria uma válvula solenoide para a irrigação. A Figura 1 (placeholder) ilustra o esquema conceitual.

4 Estudo de Caso

Para validar a proposta, executamos uma simulação do sistema com um intervalo de amostragem acelerado (10 segundos em vez de 15 minutos). Os dados do sensor foram gerados aleatoriamente dentro de faixas operacionais esperadas.

O sistema coletou 16 amostras. Em um ciclo de teste, a temperatura mínima registrada foi de 15.2°C e a máxima de 34.8°C. O sistema identificou e marcou corretamente esses dois nós para exclusão.

A média das 14 amostras restantes foi calculada. Ex:

- Média de Temperatura: 24.5°C
- Média de Umidade: 42.1%
- Média de Salinidade: 2.1 dS/m

Com base na lógica de decisão implementada (Irigar se Umidade < 45% E Salinidade < 3.0), o sistema gerou o relatório na porta Serial com a decisão: "IRRIGAR (Umidade Baixa)". O sistema então limpou a lista encadeada e iniciou o próximo ciclo de coleta, confirmando o funcionamento de todo o fluxo de dados.

5 Conclusão

Este trabalho apresentou um protótipo funcional de um sistema embarcado para coleta e análise de dados do solo. O sistema implementa com sucesso a lógica de coleta de 15 minutos, armazenamento em lista encadeada e processamento em ciclos de 4

horas, incluindo a filtragem de outliers (min/max) e a tomada de decisão autônoma.

Como melhorias futuras, sugere-se:

- Substituir a lista encadeada por um array estático (ou `std::vector`, se o compilador C++ suportar) para melhorar a robustez da memória.
- Implementar uma lógica de decisão mais sofisticada, possivelmente baseada em lógica fuzzy ou curvas de retenção de água específicas da cultura.
- Integrar comunicação sem fio (ex: LoRaWAN ou Wi-Fi) para reportar as médias e decisões a um servidor central, em vez de

apenas na porta Serial.

Referências

- [1] J. Smith, P. Jones, "IoT-Based Cloud Platform for Precision Agriculture," *Journal of Smart Systems*, vol. 5, no. 2, pp. 45-56, 2020.
- [2] A. Chen, B. Li, "Fuzzy Logic Control for Microcontroller-Based Irrigation," *IEEE Transactions on Control Systems*, vol. 12, no. 1, pp. 112-120, 2019.
- [3] R. Gupta, "Data Structures for Time-Series in Memory-Constrained Devices," *Embedded Systems Letters*, vol. 9, pp. 30-33, 2018.