

Uso do Módulo HC-SR04 para Medição de Nível da Água em Sistema Monitorado

Eros Ryan Simette
Universidade Federal do Ceará (UFC)
Sobral, Ceará, Brasil
email: erosryansimette@alu.ufc.br

Vitor Emanuel Gomes de Freitas
Universidade Federal do Ceará (UFC)
Sobral, Ceará, Brasil
email: vitoremanuel@alu.ufc.br

Danilo Victor Pacheco de Lima
Universidade Federal do Ceará (UFC)
Sobral, Ceará, Brasil
email: danilo.limalik@alu.ufc.br

Juliana de Paiva Barboza
Universidade Federal do Ceará (UFC)
Sobral, Ceará, Brasil
email: julianapaiva@alu.ufc.br

Abstract—O sensor ultrassônico HC-SR04 tem se mostrado como uma solução de baixo custo para medição de distâncias. Em conjunto com a plataforma Arduino, esse módulo permite a criação de sistemas capazes de adquirir dados com precisão considerável. Este artigo aborda sua aplicação no monitoramento de nível da água, destacando o princípio de funcionamento, a metodologia empregada para coleta de dados por meio de scripts em Python e os resultados obtidos em testes práticos, onde a altura total do reservatório foi de 21.0 cm. A medição da temperatura é essencial para aprimorar a precisão, devido à dependência da velocidade do som com a temperatura ambiente.

Index Terms—HC-SR04, sensor ultrassônico, DS18B20, Arduino, medição de nível, compensação de temperatura

ABSTRACT (ENGLISH VERSION)

The **HC-SR04** ultrasonic sensor has proven to be a low-cost solution for distance measurement. In conjunction with the **Arduino** platform, this module allows the creation of systems capable of acquiring data with considerable accuracy. This paper discusses its application in **water level** monitoring, highlighting the operating principle, the **methodology** used for data collection through Python scripts, and the **results** obtained in practical tests, where the total height of the reservoir was 21.0 cm. Temperature measurement is essential to improve accuracy, due to the dependence of the speed of sound on ambient temperature.

Keywords— HC-SR04, Arduino, Water level, Ultrasonic sensor, Temperature

I. INTRODUÇÃO

A medição e o monitoramento contínuo do nível de água são aspectos cruciais no quando tratamos dos recursos hídricos e na prevenção de inundações em sistemas urbanos de drenagem. O uso de sensores ultrassônicos de baixo custo, como o HC-SR04, oferece uma solução viável. Essa abordagem se destaca pela simplicidade de implementação e pelo potencial de integração em sistemas de **Internet das Coisas (IoT)** [1]. Adicionalmente, o protótipo incorpora o sensor de temperatura **DS18B20** [2], visando aprimorar a precisão das medições, visto que a velocidade do som varia conforme a

temperatura do ar — fator que não pode ser negligenciado em aplicações de precisão.

II. METODOLOGIA

O módulo **HC-SR04** opera com base no princípio do **sonar**: ele emite um pulso ultrassônico (pino *Trig*) e mede o tempo de retorno do eco (pino *Echo*). A distância d é calculada pela velocidade do som no ar (v) e o tempo de voo (t) [3], conforme a Equação 1.

$$d = \frac{v \cdot t}{2} \quad (1)$$

A velocidade do som (v) depende diretamente da temperatura (T) do meio, segundo a Equação 2:

$$v \approx 331.4 + 0.6 \cdot T \quad (2)$$

O **nível da água** (N) é obtido subtraindo-se a distância medida (d) da altura total do reservatório (H_{total}):

$$N = H_{total} - d \quad (3)$$

A. Montagem Experimental e Coleta de Dados

O protótipo foi construído utilizando um microcontrolador **Arduino UNO**, responsável por gerar o pulso de disparo, cronometrar o tempo de retorno e calcular a distância. O sensor HC-SR04 foi fixado na parte superior do reservatório, e o DS18B20 foi usado para medir a temperatura ambiente.

O reservatório foi criado adaptando de um pote de plástico de 6,6 L, com altura interna total de $H_{total} = 21.2$ cm. Os dados foram coletados via script Python (`coleta_serial.py`), que se comunicava com a porta serial do Arduino e salvava as medições (*tempo*, *temperatura* e *distância*) em um arquivo `.csv`.

B. Procedimento Experimental

O procedimento seguiu o roteiro a seguir:

- 1) Conectar o sensor DS18B20 ao pino digital 2 e o HC-SR04 aos pinos A0 (Trigger) e A1 (Echo) do Arduino UNO.

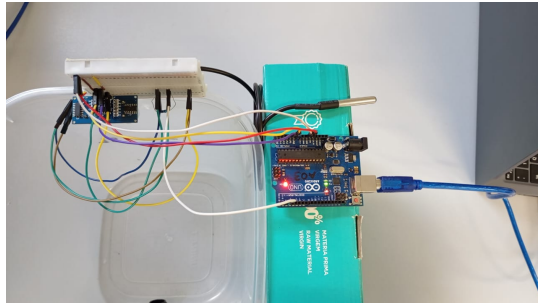


Fig. 1. Placa Arduino UNO e montagem em protoboard com HC-SR04 e DS18B20.

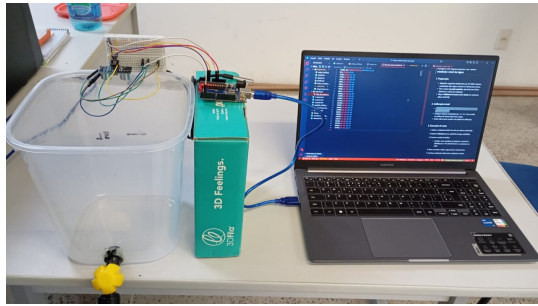


Fig. 2. Bancada de testes e sistema de controle.



Fig. 3. Reservatório utilizado nos testes experimentais.

- 2) Fazer o upload do código `coleta.ino` via Arduino IDE.
- 3) Com o recipiente vazio, executar `coleta_serial.py` e registrar as primeiras 100 amostras para medir H_{total} .
- 4) Adicionar o volume desejado de água (ex.: 3 L) e aguardar estabilização.
- 5) Abrir a torneira e coletar dados durante a drenagem.
- 6) Repetir o processo ao menos três vezes para garantir confiabilidade.

III. RESULTADOS E DISCUSSÃO

As medições foram obtidas conforme o nível de água diminuía. O script Python (`grafico.py`) processou os dados e gerou gráficos. A **Tabela I** mostra os valores extremos considerando $H_{total} = 21.2$ cm.

TABLE I
RESULTADOS DAS MEDIÇÕES DE DISTÂNCIA E NÍVEL DA ÁGUA
(BASEADO EM $H_{total} = 21.2$ cm)

Condição	Distância d (cm)	Nível N (cm)
Vazio	21.2	0.0
Intermediário	16.0	5.2
Cheio	9.9	11.3

Os resultados mostram relação linear inversa entre distância e nível. Quando vazio, o sensor registrou 21.2 cm; no estado cheio, ≈ 9.9 cm, resultando em $N = 21.2 - 9.9 = 11.3$ cm.

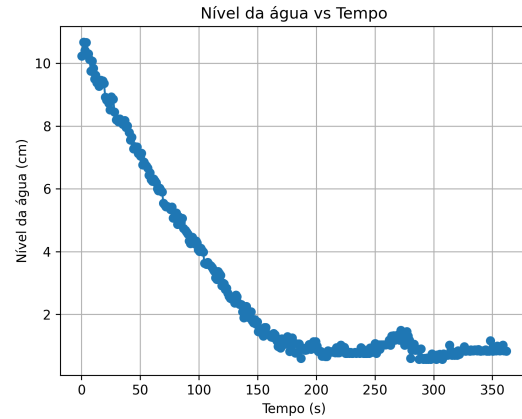


Fig. 4. Gráfico Nível da Água vs Tempo durante a drenagem.

A inclusão do DS18B20 melhora a precisão. Pequenas variações de temperatura alteram a velocidade do som [3], afetando o cálculo de d . Mesmo com temperatura estável, a medição desse parâmetro é essencial. Outro ponto a destacar é a utilização do próprio **Arduino UNO** para realizar a contagem do tempo, o que causou pequenas variações nos registros.

IV. CONCLUSÃO

O módulo ultrassônico HC-SR04, integrado ao Arduino e monitorado via Python, demonstrou ser uma ferramenta

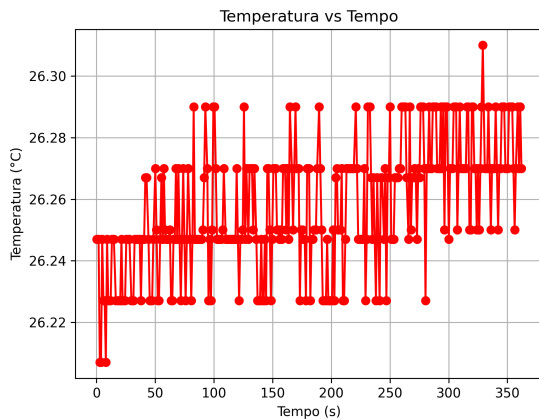


Fig. 5. Gráfico de Temperatura vs Tempo durante a coleta de dados.

eficaz e de **baixo custo** para medição de nível de água. O sistema atingiu um nível máximo de 11.3 cm no estado cheio. A metodologia baseada em coleta serial e análise via CSV oferece base sólida para estudos posteriores. Como trabalhos futuros, propõe-se implementar a **compensação térmica em tempo real** no Arduino, utilizar o módulo Real Time Clock DS1307 e integrar o sistema a uma plataforma **IoT** para monitoramento remoto de drenagens urbanas.

APPENDIX A CÓDIGOS-FONTE DO PROJETO

Os códigos utilizados neste trabalho — incluindo `coleta.ino`, `coleta_serial.py` e `grafico.py` — estão disponíveis publicamente no repositório oficial do projeto no GitHub:

https://github.com/Eros-afk/Codes_arduino_Radar_das_aguas/tree/main/Experimento_Nivel_Agua

O repositório contém:

- Código Arduino para leitura dos sensores HC-SR04 e DS18B20;
- Scripts Python para coleta e análise de dados;
- Arquivos CSV e gráficos gerados;
- Instruções detalhadas para reprodução do experimento.

O conteúdo está versionado e documentado para facilitar a replicação e futuras melhorias. O mesmo repositório também inclui outros códigos desenvolvidos no âmbito do projeto **Radar das Águas**.

REFERENCES

- [1] Arduino, “Hc-sr04 ultrasonic sensor,” 2024. [Online]. Available: <https://projecthub.arduino.cc/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-7cabe1>
- [2] Robocore, “Primeiros passos com o ds18b20,” 2025. [Online]. Available: <https://www.robocore.net/tutoriais/primeiros-passos-ds18b20-arduino>
- [3] D. Halliday, R. Resnick, and J. Walker, *Fundamentos de Física, Volume 2: Gravitação, Ondas e Termodinâmica*, 10th ed. LTC - Livros Técnicos e Científicos Editora, 2016, 17.