

Protótipo de Automação de uma Estação para Tratamento de Água da Chuva

Diego N. S. da Silva¹, Ricardo D. Marques¹, Alexandre da S. Avincola¹,
Daniela E. Flôr¹, Eduardo H. M. da Cruz¹

¹Instituto de Informática – Instituto Federal do Paraná (IFPR)

Rua José Felipe Tequinha 14.000 – 87.703-536 – Paranavaí – PR – Brazil

{nicacio16, dabroi84}@hotmail.com

{alexandre.avincola, daniela.flor, eduardo.cruz}@ifpr.edu.br

Abstract. *Free hardware prototyping has been an important tool to help researchers to develop projects in the most varied areas. Considering rainwater reuse as an important step for sustainable development and its wide availability, we propose to develop a prototype of a rainwater treatment station. It is composed of a set of hardware components, logical programming and communication between modules. We evaluated the prototype, in which results showed its efficacy.*

Resumo. *A prototipação de hardware livre tem sido uma importante ferramenta para auxiliar pesquisadores a desenvolver projetos nas mais variadas áreas. Considerando o reaproveitamento hídrico pluvial como importante passo para o desenvolvimento sustentável e sua ampla disponibilidade, propõe-se o desenvolvimento de um protótipo para tratamento de água de chuva composto por um conjunto de componentes de hardware, programação lógica, comunicação entre os módulos. O protótipo foi avaliado, demonstrando sua eficácia.*

1. Introdução

Acredita-se que diversos recursos da natureza não são amplamente utilizados pela ausência na aplicação dos recursos tecnológicos existentes. A escassez hídrica para abastecimento, seja quantitativa ou qualitativa, já é uma realidade em várias regiões do planeta. Em contrapartida, os estragos causados pelas fortes chuvas são cada vez maiores [May 2009].

A possibilidade de aproveitar água da chuva, gratuita e abundante, para diversos fins residenciais ou indústrias é globalmente reconhecida. Porém, mesmo com o crescimento exponencial na demanda do consumo hídrico, é modesto o uso de sistema de captação de água pluvial para reuso. Com o objetivo de mudar essa realidade, países como Japão, Austrália, Alemanha e República de Singapura utilizam este recurso como forma alternativa de abastecimento. Em Hamburgo (Alemanha), assim como em 16 cidades japonesas, existe um benefício pago a quem armazenar água da chuva para consumo [Colla 2008]. Entretanto, a falta de um sistema automatizado para coleta, tratamento e distribuição de água pluvial pode manter muitos usuários distantes desta realidade.

No contexto apresentado, este trabalho tem o objetivo de apresentar uma alternativa eficiente e de baixo custo para automatizar uma estação de tratamento de água da

chuva, detalhando a diferença entre os sensores ultrassônicos e sensores de nível para controlar nível de água dos reservatórios utilizados no protótipo. Sensores, reles e uma placa de desenvolvimento livre farão parte de um conjunto de hardware que identificarão a existência de água em um reservatório primário (cisterna), a necessidade de abastecimento de água no reservatório secundário (caixa d'água para consumo) e o acionamento do relé responsável pelo controle da bomba d'água quando necessário.

2. Microcontroladores

Em geral, tanto o microprocessador quanto o microcontrolador são capazes de buscar instruções na memória, executá-las, e o resultado utilizado pelos dispositivos de saída. No entanto, o primeiro se comunica com outros componentes que estão conectados através dos barramentos da placa e suporta a execução de um sistema operacional. Já o segundo é um circuito integrado que contém processador, memória e pinos de entrada e saída em um único circuito integrado, obrigando o desenvolvedor responsável pelo sistema embarcado a criar todas as rotinas necessárias para o funcionamento do dispositivo. O microcontrolador utiliza pinos de entrada e saída de dados, também chamados de GPIO (*general purpose I/O*), e dentre eles temos:

GPIO Analógica: Os dados manipulados internamente em um microcontrolador são digitais, portanto para fazer a leitura de alguns sensores, como luminosidade, pressão ou outra unidade de grandeza que possa assumir diferentes valores, o hardware precisa fazer a conversão do valor analógico para digital. Esse procedimento é realizado por um conversor analógico digital (ADC ou conversor A/D) presente nos microcontroladores. Normalmente, um único conversor é responsável pela conversão dos dados de todos os canais. O microcontrolador Arduino UNO, por exemplo, utiliza internamente um conversor A/D de 10 bits que consegue converter um sinal digital (tensão de 0V a 5V) para um valor entre 0 até 1023. Desta forma, uma tensão enviada por um sensor de temperatura é capaz de ser lida por um microcontrolador [Peixoto 2012].

GPIO Digital: Portas digitais são preparadas para analisar apenas presença ou ausência de sinal, ou seja, depois de reservado o pino para uso, faça-se a leitura dele para identificar seu estado, que pode ser ligado (0V), desligado (5V), ou até mesmo, enviando ou não sinal [Peixoto 2012].

PWM - Pulse Width Modulation: A modulação por largura de pulso consiste em manipular uma forma de onda quadrada que mantém a frequência constante e varia o intervalo de tempo que o nível lógico permanece alto durante. A razão entre o período de frequência e o tempo em que o nível lógico permanece alto é chamado *duty cycle*, ou ciclo de trabalho. A porção de tempo que permanece ligado (nível alto) é chamada de largura de pulso. Assim, é possível obter valores entre 0 e 255 a partir da modularização da largura de pulso, onde 0 corresponde a 0% e 255 a 100% do ciclo de trabalho [Barr 2001].

No início, os microcontroladores eram produzidos principalmente para atender a indústria de produtos eletrônicos, e suas atualizações visavam aumentar a capacidade de processamento para atender aplicações cada vez mais complexas. No entanto, atualmente, além da destinação acima referenciada, os microcontroladores compõem placas de desenvolvimento livre que oferecem ao usuário facilidades para desenvolver projetos com

programação e design interativo. A literatura ressalta [Alves et al. 2018] ainda o caráter didático apresentado pelos fabricantes de placas de prototipagem e a grande aceitação dos profissionais com interesse no campo das aplicações ubíquas (integração de ações e comportamentos humanos com a computação). Um relatório publicado pela Sociedade Brasileira de Computação (SBC) considera o tema com um dos grandes desafios atuais da pesquisa na área da computação [Silva et al. 2015].

A bibliografia [Neto 2014] descreve ambientes multimídia interativos como sendo aqueles em que a computação pode ser utilizada, de forma imperceptível, para melhorar as atividades do dia a dia, ou seja, com dispositivos autônomos a necessidade de ação humana para execução das atividades é reduzida. Ainda segundo o autor, pesquisas sobre computação ubíqua abordam os seguintes temas: tratamento de conteúdo multimídia, eficiência no consumo de energia, amplo acesso de dispositivos a rede sem fio, segurança, dentre outros. Acredita-se que o emprego de plataformas capazes de integrar os mais variados dispositivos em um mesmo ambiente foi fundamental para criação das placas prototipação de hardware livre. Dentre as mais utilizadas, podem-se citar três: Raspberry, Arduino e ESP. Para todas pode-se utilizar um alto nível de programação, presente nas linguagens C/C++, Python, Java, e outras [Neto 2014].

O termo Computação Ubíqua [Neto 2014] – a palavra ubíqua vem do Latim (*ubiquu*) e quer dizer “aquilo que está ao mesmo tempo em toda parte” – advém do contexto Computação Pervasiva e Computação Móvel. Na computação pervasiva, o computador é capaz de obter informações (através de sensores/serviços computacionais) do meio em que atua e, em seguida, gerenciar dinamicamente suas aplicações, de forma inteligente, para atender as necessidades do usuário. A próxima seção retrata a capacidade do usuário portar fisicamente os serviços computacionais disponíveis, isto é, o usuário dispor de um computador a todo momento. Desta forma, a computação ubíqua torna-se eficaz pela capacidade de integrar a mobilidade com a funcionalidade computacional, e qualquer dispositivo que se move com o usuário pode gerar, dinamicamente, serviços computacionais personalizados, englobando os conceitos de Internet das Coisas (IoT - Internet of Things) [Dal Berto et al.].

3. Plataformas de Desenvolvimento

Nesta seção, iremos introduzir as principais plataformas de desenvolvimento e prototipação existentes no mercado.

3.1. Arduino

Criado em 2005 por um grupo de estudantes [Alves et al. 2018], na Itália, o Arduino é um plataforma de desenvolvimento com conceito de hardware livre. Surge com objetivo didático e alternativa de baixo custo, já que na época um microcontrolador custava cerca de 100 euros. Com software capaz de transformar uma linguagem de alto nível (C++) em linguagem de máquina e uma vasta biblioteca, observa-se baixa complexidade aos usuários na implementação de comandos responsáveis pela execução das tarefas. Pode-se encontrar uma grande variedade de modelos de Arduino com diversidade de tamanhos, interfaces e quantidade de memória disponível. Custando a partir de R\$54,90¹, neste artigo, o modelo referenciado será o Arduino UNO.

¹ Preço atual disponível no site <https://www.filipeflop.com>

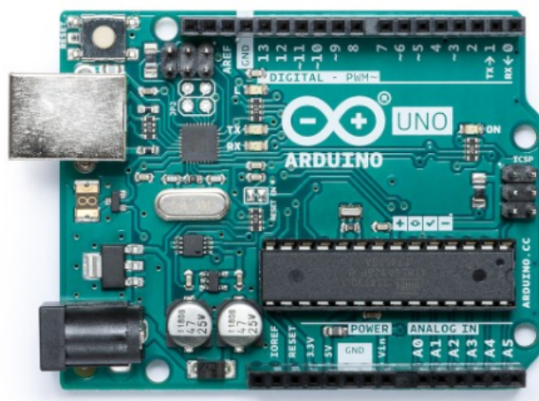


Figura 1. Arduino UNO R3. Fonte: Arduino (2019).



Figura 2. Raspberry Pi 3 Model B+. Fonte: Raspberry Pi (2019).

A Figura 1² ilustra o modelo Arduino UNO, e este contém um chip microcontrolador ATmega328P, que é composto por um CPU de 16 MHz, memória (RAM e ROM), unidade de processamento aritmético, 5 pinos de alimentação externa (GND, 5V e 3V3), um pino para reset, 14 pinos digitais de entrada e saída (GPIO Digital) e 6 pinos de entrada e saída analógica (GPIO PWM, que também pode-se utilizar como GPIO Digital) responsáveis por receber dados, processá-los e enviá-los para fontes externas. É alimentado por uma tensão em corrente contínua que pode variar de 7 á 12 Volts (suporta picos de até 20 Volts) com corrente mínima de 300 mA. Possui um regulador de tensão próprio que filtra e estabiliza a tensão de entrada para a sua operação, 5 Volts [Alves et al. 2018].

3.2. Raspberry Pi

Desenvolvido pela Fundação Raspberry Pi, o Raspberry Pi representado pela Figura 2³, possui o hardware necessário acoplado a uma única placa para ser considerado um computador com tamanho e preço reduzidos. O principal objetivo do Raspberry Pi, assim como outras placas de desenvolvimento livre, é a inserção dos conhecimentos relacionados à ciência da computação e eletrônica em placa de prototipação livre, o que possibilita a extração de resultados para fundamentar uma pesquisa ou aprimorar o protótipo.

²Imagem disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>;. Acesso em jun. 2019.

³Imagem disponível em: <https://www.raspberrypi.org/products/>;. Acesso em jun. 2019.



Figura 3. NodeMCU ESP32. Fonte: Filipeflop (2019).

O Raspberry Pi 3 Model B+ foi o modelo escolhido para apresentar o potencial deste tipo de hardware. Dotado de um processador Quad Core 64 bits com 1.4GHz e 1GB de memória, a placa é capaz de rodar sistema operacional baseado em Linux, sendo que sua distribuição oficial é o Raspbian. A placa trabalha com uma tensão (corrente contínua) de 3.3 Volts e corrente mínima de 800 mA, suportando picos de até 5 Volts. O modelo pode ser conectado a rede através da porta Ethernet ou WiFi IEEE 802.11.b/g/n/ac, ou em outros dispositivos por meio de um adaptador (integrado) Bluetooth 4.2. Dispõe também de 40 pinos GPIO, um slot para cartão microSD, saída de vídeo HDMI e 4 portas USB 2.0. É apto para trabalhar com aplicações que exigem maior poder de processamento, como por exemplo videogames, servidores e câmeras de monitoramento [de Oliveira 2017]. O valor corrente é R\$ 289,90 ⁴.

3.3. ESP

A plataforma de prototipação ilustrada na Figura 3⁵ corresponde ao NodeMCU-32S, se trata de uma evolução de hardware da família ESP, desenvolvida pela empresa Espressif. Lançado em janeiro de 2017, ela é constituída por um processador Dual Core 32 bits com clock ajustável de 80 a 240 MHz, 520 Kbytes de memória RAM, conexão WiFi 2.4 GHz (padrão 802.11 b/g/n), conexão Bluetooth 4.2 e 36 pinos GPIO. A tensão de alimentação é DC 5 Volts e corrente típica de 80 mA, opera em nível lógico com 3.3 Volts (limites 2.2V até 3.6V) e a corrente máxima por pino é de 12 mA. O valor pago pelo equipamento foi R\$ 66,90 ⁶. A Figura 6⁷ apresenta os detalhes da placa NodeMCU ESP32 [Rios et al. 2019].

Optou-se por utilizar no projeto a placa de desenvolvimento ESP32, tendo em vista o elevado desempenho oferecido pelo hardware, se comparado com Arduino UNO R3 por exemplo. Devido ao alto custo para aquisição da plataforma Raspberry Pi, o desenvolvimento do protótipo utilizando essa tecnologia tornou-se inviável. Dentre as várias características analisadas o ESP32 tornou-se vantajoso por oferecer recursos que

⁴Preço segundo <https://www.filipeflop.com/produto/raspberry-pi-3-model-b/#tab-description>

⁵Imagem disponível em: <https://www.filipeflop.com/produto/modulo-wifi-esp32-bluetooth/>;. Acesso em jun. 2019.

⁶Preço segundo <https://www.filipeflop.com/produto/esp32/>

⁷Imagem disponível em: <http://esp-wroom-32.de/>;. Acesso em jun. 2019.

atendem perfeitamente as necessidade do protótipo. A disponibilidade de comunicação via WiFi é capaz de facilitar a integração entre diversos componentes que precisam trocar dados. Outros pontos que devem ser observados são o alto desempenho para execução de tarefas, baixo consumo de energia e o baixo custo para aquisição da placa.

4. Modelagem do Protótipo

Por meio de uma ferramenta para modelagem⁸ tornou-se mais eficiente o desenho do protótipo, já que fora feito antes a identificação dos componentes eletrônicos que seriam utilizados e a disposição dos equipamentos para facilitar as conexões com a placa de hardware livre. Para que fosse possível a modelagem do protótipo e as configurações nas portas do microcontrolador, utilizou-se o *PINOUT* (Figura 6) representativo da placa. Para aferir o nível de água dos reservatórios e manter-se dentro do escopo deste trabalho foram identificados dois componentes tecnológicos facilmente encontrados no mercado e com baixo custo para aquisição. A escalabilidade do protótipo também foi considerada durante a escolha dos componentes, desta forma, os conceitos e tecnologias discutidos neste artigo são capazes de ser aplicados em indústrias, comércio e residências contribuindo para o desenvolvimento de soluções inovadoras.

As Figuras 4 e 5 mostram os componentes empregados na montagem do protótipo. Para facilitar a conexão dos componentes eletrônicos, utilizou-se uma protoboard (placa de ensaio com linhas e colunas condutoras de energia) e jumpers (fio de cobre com pontas macho/fêmea) que facilitam as ligações entre os equipamentos. Para indicar os níveis de água do primeiro reservatório (caixa d'água predial), empregou-se LEDs, sendo vermelho para nível baixo, amarelo para intermediário e verde para cheio. O mesmo ocorreu com o segundo reservatório (cisterna). Já o LED azul foi utilizado como indicador de que o relé está ativo. Foi adotado como solução para não danificar os LEDs do sistema limitar a corrente e oferecer um queda de tensão através de um resistor de 220 Ohms (para cada LED).

4.1. Sensor de Nível

Inicialmente foram utilizados nos testes dois sensores de nível por reservatório, que eram responsáveis por identificar a lâmina d'água com nível baixo ou alto de cada recipiente. Para implementação deste componente no protótipo (através da protoboard), foi necessário conectar um dos pinos do sensor na porta VIN do ESP32, e o outro na porta GND (passando por um resistor de corrente). O microcontrolador, através de uma das suas portas (previamente definida), consegue captar quando o sinal do sensor está fechado ou aberto, ou seja, quando o sinal estiver fechado significa que a lâmina d'água está acima do sensor (dependendo da posição que for instalada no reservatório de água) e conseqüentemente existe água, caso contrário o sensor permanecerá com sinal aberto (ausência de água).

O sensor de nível utilizado na Figura 4 mostrou-se eficiente ao que foi proposto, apresentando leituras consistentes durante todo período de teste. Entretanto, a ausência na leitura dos dados no momento em que a lâmina d'água está fora da faixa de leitura dos sensores dificulta a informação a cerca da autonomia do sistema de abastecimento. Por

⁸Adotou-se o software Fritzing, open-source e multiplataforma, para executar a modelagem do protótipo.

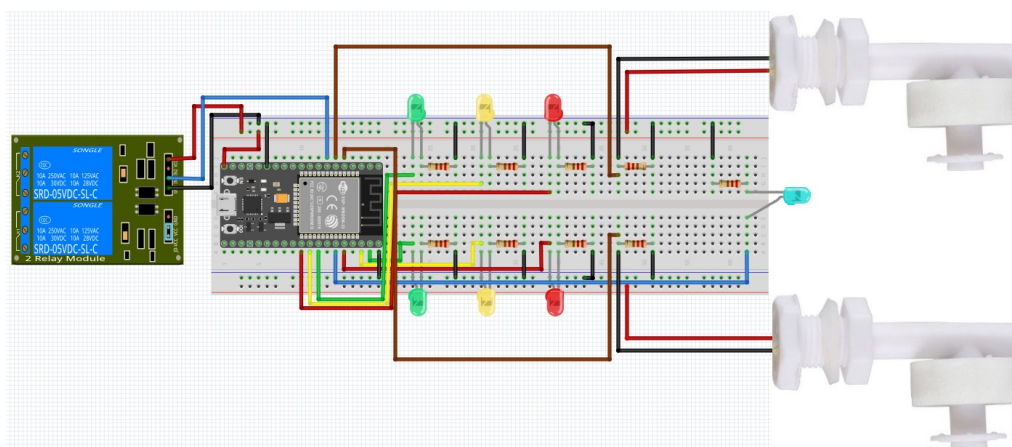


Figura 4. Modelagem do protótipo utilizando sensor de nível.

exemplo, quando os recipientes possuem fluxo de água (esvaziando/enchendo) e o nível hídrico está entre o nível mínimo e nível máximo, o sistema continua funcionando, mas o usuário não tem acesso aos indicadores a cerca da quantidade de água disponível para abastecimento.

4.2. Sensor Ultrassônico

Na Figura 5, observa-se que foram implementados sensores ultrassônicos no protótipo para a segunda fase de testes, isso possibilitou comparar as duas tecnologias para aferição de nível d'água citadas neste artigo e ter respaldo científico para determinar qual melhor opção para resolver o problema em questão. A escalabilidade do protótipo é um fator importante, pois pode propiciar o emprego das tecnologias discutidas no decorrer desse trabalho em um cenário real.

O conceito aplicado pelo sensor ultrassônico é que uma onda sonora de alta frequência emitida pode refletir nos obstáculos e voltar para o sensor. Sempre que isso ocorre tem-se o tempo gasto por cada pulso ultrassônico para percorrer a distância entre a ida e volta. Com base nessas informações, o microcontrolador consegue calcular e retornar a distância entre o sensor e a lâmina d'água. Os morcegos utilizam-se desta técnica – chamada ecolocalização – para detectar a posição dos objetos e assim conseguir voar e caçar [Stoppa 2012].

Após a definição dos pinos Echo e Trigger, receptor do pulso e emissor do pulso ultrassônico respectivamente, no código, foram feitas as conexões entre os componentes, onde o GND do sensor estará conectado à porta respectiva do ESP32, o pino VCC ligado ao VIN do microcontrolador e os pinos echo e trigger nas portas definidas do código. Ao contrário do sensor utilizado na Figura 4, este é capaz de fornecer valores que variam entre 2 cm e 400 cm (de acordo com a lâmina d'água) e com isso extrair informações relevantes á solução problema. Os resultados dos testes foram positivos e o sensor ultrassônico portou-se como uma alternativa eficiente para identificação de nível d'água em reservatórios, mesmo durante o período em que os reservatórios estão sendo alimentados ou esvaziados.

Em consenso com o escopo do trabalho que visa uma solução eficiente e com baixo custo de implementação para automatizar uma estação de tratamento de água da

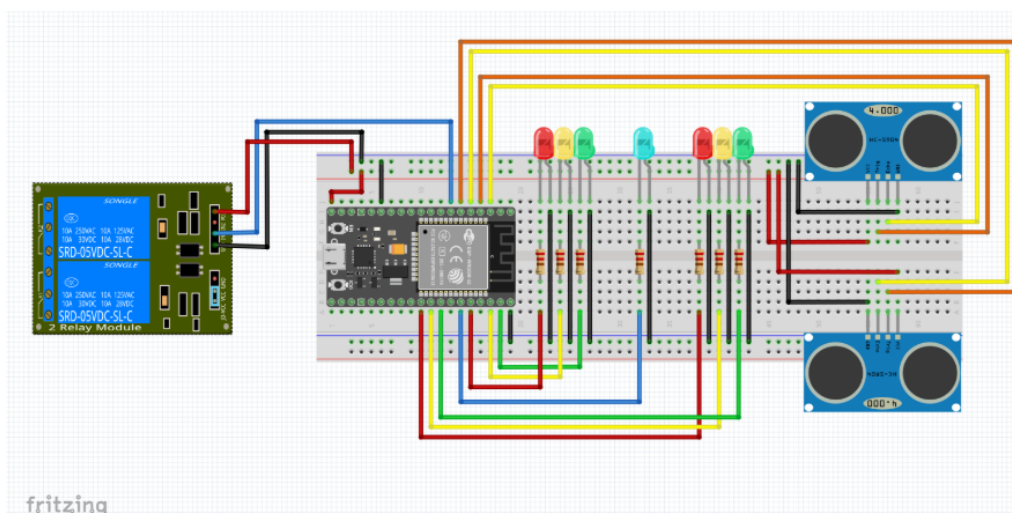


Figura 5. Modelagem do protótipo utilizando sensor ultrassônico.

chuva, decidiu-se empregar o sensor ultrassônico ao protótipo já que os resultados dos testes foram satisfatórios.

4.3. Relé

Para acionar um dispositivo externo, como uma bomba d'água por exemplo, que trabalha com tensão e corrente diferentes das praticadas dentro do sistema eletrônico, foi necessário o uso de um relé. Desta forma, quando o microcontrolador emite um sinal lógico para acionar o relé, ele altera seu estado (ligado ou desligado) o que permite ligar ou desligar o dispositivo conectado ao relé. Através deste sinal, gera-se um campo magnético na bobina fazendo com que os contatos elétricos permitam a passagem de energia.

Outro dispositivo que pode trabalhar como uma espécie de interruptor digital (para equipamentos com baixa corrente elétrica) é o transistor, composto por três terminais: emissor, base e coletor. Onde emissor e coletor são conectados a um circuito elétrico em série que, alimentados ou não por uma corrente, e a base (conectada a outro ponto do circuito) é responsável por controlar o estado do transistor. Por isso, quando aplica-se uma tensão sobre a base, o transistor se comporta como condutor e permite a passagem de corrente elétrica entre emissor e coletor alimentando o circuito. Quando for nula a tensão aplicada a base, o transistor torna-se isolante e impede o fluxo de corrente [Marques 1996].

5. Ambiente de Desenvolvimento

Empregou-se como principal ferramenta de desenvolvimento a IDE Arduino, onde foi permitido operar em um ambiente com interface gráfica que possui um compilador GCC (GNU Compiler Collection) para várias linguagens, dentre elas C e C++. Com ela foi possível escrever o algoritmo do programa, adicionar bibliotecas, compilar e enviar o código para o ESP32, e usando a interface serial foi possível acompanhar sua execução.

A biblioteca `Ultrasonic.h` utilizada traz consigo regras e funções para manipulação do sensor ultrassônico junto ao microcontrolador simplificando a implementação. Após a definição dos pinos da placa de prototipação de hardware livre

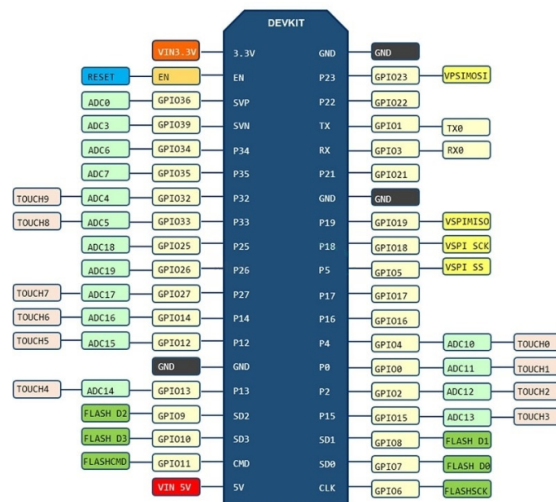


Figura 6. PINOUT ESP32. Fonte: ESP-WROOM-32 (2019).

que devem controlar os sinais do sensor ultrassônico (Figura 7), estaciona-se um objeto para cada sensor passando como parâmetro o *Trigger* e *Echo* que por fim irão retornar a distância (centímetros) entre o sensor e a lâmina d'água.

A função descrita na Figura 9 define a lógica implementada para o acionamento do relé. A condição utiliza as funções `getDistanceCaixa()` e `getDistanceCisterna()` para retornar a distância entre o sensor e o reservatório.

```
1 #include <Ultrasonic.h>
2
3 #define PIN_ECHO_CAIXA 2
4 #define PIN_TRIGGER_CAIXA 3
5
6 #define PIN_ECHO_CISTERNA 4
7 #define PIN_TRIGGER_CISTERNA 5
8
9 Ultrasonic ultrasonicCAIXA(PIN_TRIGGER_CAIXA, PIN_ECHO_CAIXA);
10 Ultrasonic ultrasonicCISTERNA(PIN_TRIGGER_CISTERNA,
    PIN_ECHO_CISTERNA);
```

Figura 7. Implementação da Biblioteca Ultrasonic.

```
1 void setup()  
2 {  
3     Serial.begin(115200);  
4     configurarPinos();  
5     Serial.println("Iniciando o sistema");  
6     Serial.println("Lendo dados do sensor");  
7 }
```

Figura 8. Função Setup.

```

1 void acionaRele()
2 {
3     if (getDistanceCaixa() > 30 && getDistanceCisterna() < 30)
4     {
5         do{
6             digitalWrite(PIN_RELE_BOMBA, HIGH);
7             digitalWrite(LED_BOMBA_LIGADA, HIGH);
8             Serial.println("BOMBA LIGADA");
9         }while (getDistanceCaixa() > 5 && getDistanceCisterna() <
10             30);
11     }
12     else if (getDistanceCaixa() < 5 || getDistanceCisterna() > 30)
13     {
14         digitalWrite(PIN_RELE_BOMBA, LOW);
15         digitalWrite(LED_BOMBA_LIGADA, LOW);
16         Serial.println("bomba desligada");
17     }
18 }

```

Figura 9. Função de Acionamento do Relé.

Assim, caso o espaço aferido pelo sensor ultrassônico da caixa d'água seja superior a 30 cm, ou seja o recipiente está com baixo volume, e a distância da cisterna seja inferior a 30 (existe água no reservatório) a bomba d'água será ligada (escreve *HIGH* na porta digital correspondente), assim como o LED, por fim ainda deverá aparecer uma mensagem no serial monitor informando o estado da bomba.

6. Conclusão e Trabalhos Futuros

Em consonância com a revisão literária, observou-se que os componentes de hardware utilizados para montagem do protótipo e o algoritmo desenvolvido para o controlar o sistema responderam satisfatoriamente aos testes realizados, destacando a utilização do sensor ultrassônico no protótipo, já que durante os testes notou-se facilidade de adaptação nos mais diferentes tipos e modelos de reservatórios. Do mesmo modo, percebeu-se o potencial didático que o emprego das plataformas de desenvolvimento podem gerar no meio acadêmico, onde durante a construção do protótipo tiveram-se aplicados diferentes conhecimentos de eletrônica, algoritmos e lógica de programação.

Em pesquisas futuras, acrescenta-se a necessidade de implementar os conceitos discutidos neste artigo em um protótipo funcional para uma estação de tratamento de água da chuva, com a perspectiva de fazer análises físico-químicas para constatar a eficiência dos filtros de partículas sólidas em suspensão na água e a eficácia no tratamento microbiológico utilizando a tecnologia de desinfecção ultra violeta. Pretende-se também interligar o protótipo micro-controlado ao sistema de filtração e tratamento de água para analisar a efetividade no dimensionamento da bomba d'água de acordo com a autonomia de filtragem e aplicar conceitos de Internet das Coisas, possibilitando a interação web do usuário com o sistema.

Referências

- Alves, E. S. B. C., Brod, F. A. T., and Da Silva, M. A. B. V. (2018). Robótica educacional livre no ensino de física: experimentação em termodinâmica.
- Barr, M. (2001). Pulse width modulation. *Embedded Systems Programming*, 14(10):103–104.
- Colla, L. L. (2008). Sistemas de captação e aproveitamento de água de chuva.
- Dal Berto, J. C., Cagliari, A. I., and Colling, J. Automação residencial: Protótipo de janela automatizada com arduíno residential automation: Automated window prototype with arduin.
- de Oliveira, S. (2017). *Internet das coisas com ESP8266, Arduino e Raspberry PI*. Novatec Editora.
- Marques, A. E. B. (1996). *Dispositivos semicondutores: diodos e transistores*. Saraiva Educação SA.
- May, S. (2009). *Caracterização, tratamento e reúso de águas cinzas e aproveitamento de águas pluviais em edificações*. PhD thesis, Universidade de São Paulo.
- Neto, M. C. M. (2014). Desenvolvimento de aplicações ubíquas com ar-duino e raspbery pi. *Minicursos do XX Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 1–30.
- Peixoto, R. L. (2012). Monitoramento da descarga de bateria com o uso de microprocessador arm. *Universidade Federal do Rio de Janeiro, Ria de Janeiro, Tese Licenciatura*.
- Rios, A. d. A. et al. (2019). Aplicação de internet das coisas no monitoramento de corrente, tensão e temperatura em motor de indução trifásico.
- Silva, E., Botelho, L., Santos, I., and Sanchez, G. (2015). Computação ubíqua—definição e exemplos. *Revista de Empreendedorismo, Inovação e Tecnologia*, 2(1):23–32.
- Stoppa, M. H. (2012). A robótica educacional em experimentos elementares de física. *Instrumento-Revista de Estudo e Pesquisa em Educação*, 14(1).