



ROBÓTICA INCLUSIVA · 2024-2 · Relatório Projeto Final

Sistema de Monitoramento e Controle Hidráulico Inteligente para Irrigação automática com Arduino

Autores

- Rafael Sampaio e Silva e Tarsila Amado Alves de Brito

Resumo

Este projeto visa desenvolver um sistema de vaso autoirrigável com o objetivo de simplificar o cuidado de plantas para pessoas com rotinas intensas e idosos com Alzheimer, promovendo o bem-estar através do contato com a natureza. O sistema utiliza um robô equipado com o Sensor de Umidade de Solo FC-28 para monitorar o nível de umidade do solo e uma Mini Bomba de Água 3-6v RS-385 para irrigação automática. A inovação está na possibilidade de configuração da frequência de irrigação, permitindo que cada planta receba cuidados personalizados, mesmo em ausência de supervisão constante, otimizando o uso de água e garantindo a saúde da planta.

Palavras-chave: Automatização, Monitoramento, Autossustentável, Autoirrigável.

1 Introdução

A presença de plantas em ambientes internos está associada a benefícios físicos e psicológicos, especialmente para idosos com Alzheimer, pois o contato com a natureza auxilia na estimulação sensorial e emocional. No entanto, cuidar de plantas pode ser um desafio para indivíduos com dificuldades de mobilidade, memória ou tempo limitado para rega constante. Visando oferecer uma solução prática, este projeto propõe o desenvolvimento de um vaso autoirrigável que monitora o nível de umidade do solo e realiza a irrigação de forma autônoma e personalizada. O sistema permite ajustar a frequência de irrigação, adaptando-se às necessidades específicas de diferentes espécies de plantas e proporcionando facilidade e confiabilidade no cuidado.

2 Descrição do Experimento

Para a implementação do vaso autoirrigável, utilizou-se um Sensor de Umidade de Solo FC-28, responsável por mensurar o nível de umidade da terra. A leitura do sensor é processada por um controlador, que, conforme a configuração estabelecida pelo usuário, ativa uma Mini Bomba de Água 3-6V RS-385 para irrigação. O sistema permite configurar frequências de irrigação específicas para diferentes tipos de plantas, garantindo que a irrigação ocorra apenas no momento adequado, mesmo quando o sensor detecta baixa umidade. Dessa forma, o usuário pode escolher os intervalos de rega ideais para cada planta, garantindo a sua saúde e otimizando o consumo de água.

3 Materiais e Versões

3.1 Sistema Versão 1

A primeira versão do sistema de vaso autoirrigável foi desenvolvida em uma simulação online(ThinkerCAD) com o objetivo inicial de criar uma irrigação automática básica. Esta versão utilizava o Sensor de Umidade de Solo FC-28 para medir a umidade do solo, porém, a frequência de irrigação não era configurável e a bomba de água ainda não tinha sido adicionada ao projeto. Apesar de fornecer uma base funcional, esta versão era muito incompleta, levando à necessidade de aprimoramentos.

3.2 Sistema Versão 2

Na segunda versão, o sistema foi atualizado para incluir controle de frequência de irrigação, permitindo ao usuário ajustar entre três opções: Frequência Baixa, Média e a Alta. Em cada uma dessas frequências, foi delimitado o tempo da irrigação(ou seja, quantidade da água usada para regar a planta), abordando o nosso objetivo do cuidado de diferentes tipos de plantas. Apesar disso, a Mini Bomba de Água 3-6V RS-385 ainda não foi incorporada ao sistema, pela necessidade futura de cuidados na infraestrutura do software e testes dos componentes eletrônicos em condições reais(*Resultados – Experimento 3*). Por fim, a simulação online dessa versão foi adicionado um servo motor temporário para simular a abertura de uma válvula, ou acionamento da bomba que será usada futuramente.

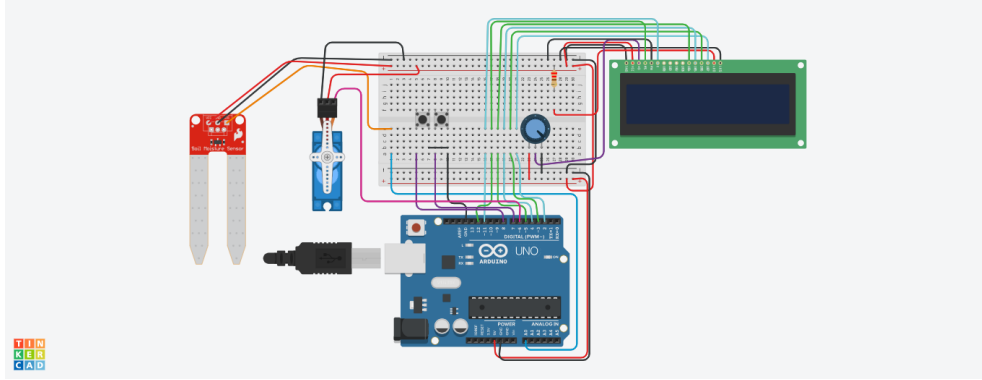


Figura 1: Circuito estruturado no Tinkercad

3.3 Sistema Versão 3

Na terceira versão, foi onde o sistema foi posto em prática com experimentos(*Resultados – Experimento 2*) feitos em sala para teste dos componentes em condições reais. Depois dos testes, foi necessário a alteração e aprimoramento do algoritmo, houve a inclusão de um temporizador e a alteração do recurso de frequência, que mudou para Alta, Diária e a Cada 2 Dias. Além disso, finalmente a Mini Bomba de Água 3-6V RS-385 foi incorporada ao sistema, acionada automaticamente pelo sensor de umidade quando o nível de umidade estiver abaixo de 40%, ou se a frequência escolhida entrar nas condições necessárias.

3.4 Algoritmos

Listing 1: Código Versão 1

```
1 #include <LiquidCrystal.h>
2
3 #define umidadeAnalog A0
4 #define pushbuttonOptions 7
5 #define pushbuttonDone 8
6
7 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
8
9 int humidityValue;
10 int nivelDeIrigacao = 1;
11 bool configurandoNivel = false;
12 bool seJaRegou = false;
13 bool sePodeRegar = false;
14 int lastButtonStateOptions = HIGH;
15 int lastButtonStateDone = HIGH;
16
17 void MenuLCD(){
18     lcd.clear();
19
20     lcd.setCursor(0, 0);
21     lcd.print("Humidade: ");
```

```

22  lcd.print(humidityValue);
23
24  lcd.setCursor(0, 1);
25
26  // Exibir a frequencia de irrigacao selecionada
27  if(nivelDeIrigacao == 1){
28      lcd.print("Freq: BAIXA");
29  }else
30  if(nivelDeIrigacao == 2){
31      lcd.print("Freq: MEDIA");
32  }else
33  if(nivelDeIrigacao == 3){
34      lcd.print("Freq: ALTA");
35  }
36  }
37
38  void ConfigurandoMenu() {
39      lcd.clear();
40      lcd.setCursor(0, 0);
41      lcd.print("Usar Frequen:");
42
43      lcd.setCursor(0, 1);
44
45      // Mostrar a frequencia atual
46      if(nivelDeIrigacao == 1){
47          lcd.print("---->BAIXA");
48
49      }else
50      if(nivelDeIrigacao == 2){
51          lcd.print("---->MEDIA");
52      } else
53      if(nivelDeIrigacao == 3){
54          lcd.print("---->ALTA");
55      }
56  }
57
58  void abrirValvula(){
59  }
60  void fecharValvula(){
61  }
62
63  void setup(){
64      pinMode(umidadeAnalog, INPUT);
65      pinMode(pushbuttonOptions, INPUT_PULLUP);
66      pinMode(pushbuttonDone, INPUT_PULLUP);
67
68      Serial.begin(9600);
69      lcd.begin(16, 2);
70  }
71
72  void loop() {
73      fecharValvula();
74
75      humidityValue = analogRead(umidadeAnalog);
76      humidityValue = map(humidityValue, 1023, 315, 0, 100);
77      Serial.print("Humidity: ");
78      Serial.print(humidityValue);
79
80      int buttonStateOptions = digitalRead(pushbuttonOptions);
81      if(buttonStateOptions == LOW && lastButtonStateOptions == HIGH){
82          configurandoNivel = !configurandoNivel;

```

```

83     delay(200);
84 }
85 lastButtonStateOptions = buttonStateOptions;
86
87 if(configurandoNivel){
88     ConfigurandoMenu();
89
90     int buttonStateDone = digitalRead(pushbuttonDone);
91
92     if(buttonStateDone == LOW && lastButtonStateDone == HIGH){
93         nivelDeIrigacao++;
94         if(nivelDeIrigacao > 3){
95             nivelDeIrigacao = 1; // Resetar para BAIXA
96         }
97         delay(200); // Debounce
98     }
99     lastButtonStateDone = buttonStateDone;
100 }
101 else{
102     MenuLCD();
103 }
104 delay(100);
105
106 humidityValue = analogRead(umidadeAnalog);
107 humidityValue = map(humidityValue, 1023, 315, 0, 100);
108 if(humidityValue <= 40){
109     abrirValvula();
110     delay(600);
111     fecharValvula();
112     delay(2000);
113 } else if(humidityValue <= 70){ // Ja esta Humido
114     fecharValvula();
115 }
116 }

```

Listing 2: Código Versão 3 – Final

```

1  #include <LiquidCrystal.h>
2
3  #define umidadeAnalog A0
4  #define pushbuttonOptions 7
5  #define pushbuttonDone 8
6  #define relePin 9
7
8  LiquidCrystal lcd(2, 3, 4, 5, 11, 12);
9
10 int humidityValue;
11 int nivelDeIrigacao = 1;
12 bool configurandoNivel = false;
13 unsigned long ultimoTempoDeIrigacao = 0;
14 unsigned long intervaloDeIrigacao = 0;
15
16 // Definir os intervalos em milissegundos
17 #define INTERVALO_DIARIO 86400000
18 #define INTERVALO_2DIAS 172800000
19
20 void MenuLCD(){
21     lcd.clear();
22     lcd.setCursor(0, 0);
23     lcd.print("Humidade: ");
24     lcd.print(humidityValue);

```

```

25  lcd.setCursor(0, 1);
26
27  if(nivelDeIrrigacao == 1){
28      lcd.print("Frequencia: ALTA");
29  }else
30  if(nivelDeIrrigacao == 2){
31      lcd.print("Frequencia:DIARIA");
32  }else
33  if(nivelDeIrrigacao == 3){
34      lcd.print("Frequencia:2DIAS");
35  }
36  }
37  void ConfigurandoMenu(){
38      lcd.clear();
39      lcd.setCursor(0, 0);
40      lcd.print("Usar Frequencia:");
41      lcd.setCursor(0, 1);
42
43      if (nivelDeIrrigacao == 1){
44          lcd.print("---->ALTA");
45      }else
46      if(nivelDeIrrigacao == 2){
47          lcd.print("---->DIARIA");
48      }else
49      if(nivelDeIrrigacao == 3){
50          lcd.print("---->2 DIAS");
51      }
52  }
53  void abrirValvula(){
54      digitalWrite(relePin, HIGH);
55  }
56  void fecharValvula(){
57      digitalWrite(relePin, LOW);
58  }
59
60  void abrirBombaRapido() {
61      digitalWrite(relePin, LOW);
62      unsigned long tempoBomba = millis();
63      while (millis() - tempoBomba < 5000) {
64
65          int buttonStateOptions = digitalRead(pushbuttonOptions);
66          int buttonStateDone = digitalRead(pushbuttonDone);
67
68          if (buttonStateOptions == LOW || buttonStateDone == LOW) {
69
70              configurandoNivel = true;
71              fecharValvula();
72              return;
73          }
74          delay(100);
75      }
76      digitalWrite(relePin, HIGH);
77      delay(1000);
78  }
79
80  void loop() {
81      humidityValue = analogRead(umidadeAnalog);
82      humidityValue = map(humidityValue, 1023, 315, 0, 100);
83
84      // Leitura e alternancia do estado dos botoes
85      int buttonStateOptions = digitalRead(pushbuttonOptions);

```

```

86  static int lastButtonStateOptions = HIGH;
87  if (buttonStateOptions == LOW && lastButtonStateOptions == HIGH) {
88      configurandoNivel = !configurandoNivel;
89      delay(200);
90  }
91  lastButtonStateOptions = buttonStateOptions;
92
93  // Modo configuracao
94  if (configurandoNivel) {
95      ConfigurandoMenu();
96
97      int buttonStateDone = digitalRead(pushbuttonDone);
98      static int lastButtonStateDone = HIGH;
99      if (buttonStateDone == LOW && lastButtonStateDone == HIGH) {
100          nivelDeIrrigacao++;
101          if (nivelDeIrrigacao > 3) {
102              nivelDeIrrigacao = 1;
103          }
104          delay(200); // Debounce
105      }
106      lastButtonStateDone = buttonStateDone;
107
108  } else {
109      // Exibe menu LCD no modo padrao
110      MenuLCD();
111
112      unsigned long tempoAtual = millis();
113
114      // Logica de irrigacao =====
115      // Modo ALTA
116      if (nivelDeIrrigacao == 1) {
117          if (humidityValue < 40) {
118              abrirBombaRapido();
119          }
120          // Modo DIARIA
121      } else if (nivelDeIrrigacao == 2) {
122          intervaloDeIrrigacao = INTERVALO_DIARIO;
123          if ((tempoAtual - ultimoTempoDeIrrigacao >= intervaloDeIrrigacao)
124              && humidityValue < 40) {
125              abrirBombaRapido();
126              ultimoTempoDeIrrigacao = tempoAtual;
127          }
128      } else if (nivelDeIrrigacao == 3) { // Modo CADA 2 DIAS
129          intervaloDeIrrigacao = INTERVALO_2DIAS;
130          if ((tempoAtual - ultimoTempoDeIrrigacao >= intervaloDeIrrigacao)
131              && humidityValue < 40) {
132              abrirBombaRapido();
133              ultimoTempoDeIrrigacao = tempoAtual;
134          }
135      }
136  }
137
138  delay(100);
139 }

```

3.5 Metodologia, estrutura e funcionamento

Durante a construção do protótipo do robô, foi adotada a Metodologia de Engenharia de Sistemas em V. Esse modelo se adequa para o projeto em questão pois é usado em projetos de hardware e software. Ele organiza o trabalho em etapas sequenciais, mas permite iterações e testes antes de avançar. Dessa maneira, o trabalho foi

dividido em 3 etapas:

- Definição e planejamento (fase 1): Projetar o sistema no Tinkercad, com simulação.
- Prototipagem e testes preliminares (fase 2): Fase de construção do circuito inicial, desenvolvimento do algoritmo e identificação de erros.
- Integração e testes finais (fase 3): Construção do protótipo final, onde todos os componentes são integrados.

Como protótipo, foi construída a “estação de irrigação”. O robô possui aproximadamente 40 cm de altura, 30 cm de largura e 30 cm de comprimento. Ele foi projetado com uma divisão funcional em duas seções principais: a metade inferior abriga o reservatório de água, responsável por armazenar o líquido utilizado na irrigação das plantas, do qual parte um pequeno cano(25mm)conectado ao vaso da planta. A metade superior concentra os componentes eletrônicos do sistema, incluindo o Arduino, os botões de controle do display LCD e, na parte superior, o sensor de umidade, que se conecta diretamente ao solo do vaso da planta.

Lista de Materiais:

- Arduino Uno
- PCB Protoboard
- Visor o LCD 16x2
- Cabos Jumpers
- Mini Bomba d'água 120l, 3v a 6v
- Diffuser lm393
- Sensor de umidade de solo
- Modulo Relé MPA-S-105-C
- Potenciometro 10k
- 2x Resistor 220 Oms
- 2x Microswitch Push Button 4 Pinos
- Conector de bateria 9v
- Bateria 9v
- Contector 4 pilhas AAA 1,5v
- 4 pilhas AAA 1,5v
- 2x recipientes de plástico para bolo
- Cano de aquario 25mm diâmetros
- Fita isolante

3.6 Fluxograma

[Figura 2, ilustrada na próxima página]

4 Resultados

4.1 Resultados – Experimento 1(sistema na versão 2)

O experimento 1 teve como o objetivo testar o funcionamento do sensor de

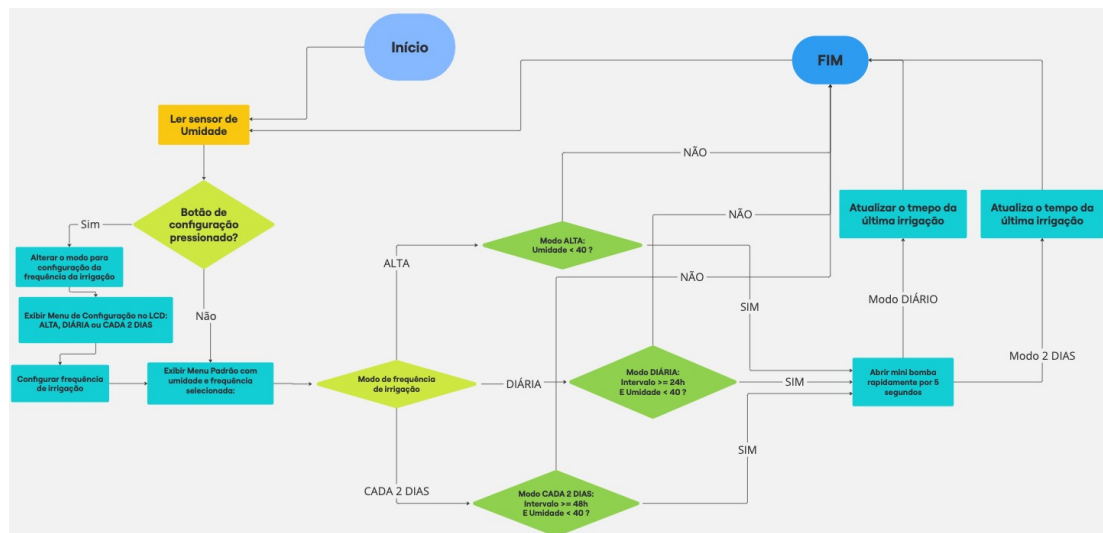


Figura 2: Fluxograma do código final

4.2 Resultados – Experimento 2(sistema na versão 3)

O experimento 2 foi planejado para conseguir os valores necessários da umidade do solo em diferentes estados. Durante o experimento em sala de aula, tivemos 4 recipientes cada um com terra em diferentes estados de irrigação: Primeiro recipiente nunca foi regado, segundo foi regado nas últimas 48 horas, terceiro foi regado nas últimas 24 horas, quarto foi regado minutos antes da medição do sensor. Dessa maneira, com os dados coletados foi observada a necessidade de alterações algoritmo .

Após a implementação do código e a montagem do sistema de irrigação padrão com os componentes especificados, o sistema de irrigação automática apresentou um funcionamento bem-sucedido. O sensor de umidade de solo FC-28 detecta o nível de umidade do solo e aciona a Mini Bomba de Água 12V RS-385 conforme as condições preestabelecidas, permitindo irrigação personalizada para diferentes tipos de plantas.

O sensor de umidade opera de acordo com os seguintes casos:

- Caso 1: Quando o nível de umidade é inferior ou igual a 40%, o sistema ativa a irrigação automaticamente.
- Caso 2: Quando o nível de umidade é superior a 40%, a irrigação não é acionada(depende da frequência estabelecida).

Esses valores foram determinados para garantir que a planta receba água apenas quando necessário, evitando a irrigação excessiva. Na tela de controle, o usuário pode configurar a frequência de irrigação com três opções:

- Frequência Alta: irrigação ocorre várias vezes ao dia.
- Diária: irrigação ocorre uma vez por dia(24 horas).
- A cada 2 Dias: irrigação é realizada a cada dois dias(72 horas).

Os testes demonstraram que o sistema responde de maneira eficiente às configurações de frequência escolhidas pelo usuário, proporcionando flexibilidade no cuidado com plantas e promovendo o uso otimizado de água.

5 Considerações Finais

O desenvolvimento deste vaso autoirrigável alcançou o objetivo de simplificar o cuidado com plantas, oferecendo uma solução prática e eficiente para pessoas com rotinas ocupadas e para idosos, especialmente aqueles com Alzheimer, que podem se beneficiar do contato com a natureza. O sistema demonstrou a capacidade de monitorar a umidade do solo e realizar a irrigação de forma automatizada e personalizada, com configurações de frequência ajustáveis pelo usuário.

Os testes mostraram que o sistema responde bem às variações de umidade e às configurações de frequência, promovendo a hidratação ideal das plantas e evitando tanto a irrigação excessiva quanto a insuficiente. Com o ajuste de três níveis de frequência (alta, diária e a cada dois dias), o sistema se adapta às necessidades específicas de diferentes tipos de plantas e facilita o cuidado, tornando-o mais acessível.

Este projeto representa um passo importante na aplicação de tecnologias simples e acessíveis para melhorar a qualidade de vida e promover o bem-estar através da natureza. Em projetos futuros, o sistema pode ser expandido com sensores adicionais e conectividade remota, permitindo monitoramento e controle via dispositivos móveis e tornando a solução ainda mais robusta e prática para o usuário.