

**PROJETO E IMPLEMENTAÇÃO DE  
UM SISTEMA DE EMBARCADO DE  
REDE INTERATIVA PARA  
MONITORAMENTO E CONTROLE DE  
UMIDADE E TEMPERATURA PARA  
USO EM ESTUFA**

Adrielle Maria Alves Queiroz

# OBJETIVO

- Supervisionar e ajustar a temperatura e a umidade em estufas
- Empregar um sistema embarcado integrado a um computador, proporcionando uma interface de fácil utilização

# PROBLEMAS

- Restrições dos sistemas convencionais que utilizam telas de LCD e botões
- Exige supervisão e controle

# SOLUÇÃO

- Sistema embarcado que se comunica com um computador por meio de portas COM
- O sistema utiliza o monitor e o teclado do computador para proporcionar uma interação mais eficiente para o usuário
- Possui capacidade de armazenamento de dados, permitindo uma recuperação rápida e conveniente
- O monitoramento e controle podem ser feitos remotamente



# COMPONENTES

## Placa de Monitoramento

- Sensores de temperatura (LM35) e umidade (LM324SG)
- ADC0809 via porta COM1
- Microcontrolador Atmel 89S52

## Placa de Controle

- CI drive ULN2003 para acionamento com RS232
- Conexão via porta COM2
- Conector MAX232 para compatibilidade com RS232

## Servidor Embarcado

- Conectado á LAN
- Conexão até 100 nós em topologia STAR



# Interface

- Exibe os valores da temperatura e umidade
- Botões para atualizar limites e gerar relatórios
- Exibição contínua do status dos dispositivos de controle

## Como funciona o Sistema

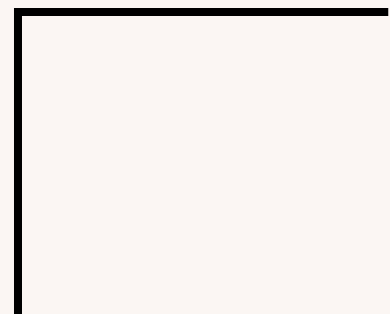
- Sensores captam valores de temperatura e umidade
- Microcontrolador que calibra e envia dados para o computador
- ADC onde converte os valores analógicos para digitais
- Interface que exibe valores e permite ajuste
- Sistema aciona automaticamente dispositivos de controle conforme os limites definidos

## Desempenho do Sistema

- A exatidão do monitoramento está diretamente ligada à calibração dos sensores
- O sistema é seguro e oferece um custo-benefício excelente.
- Uma rede LAN facilita o monitoramento e o controle remoto de maneira eficaz.

**CONCLUSÃO**

- O sistema é uma solução eficaz e econômica para monitoramento e controle em estufas
- Hardware e Software disponíveis torna o sistema acessível



**MELHORIAS A  
FAZER  
FUTURAMENTE**

- Aumentar a exatidão com a utilização de microprocessadores de 16 bits e sensores mais avançados.
- Ampliar o controle para incluir outros parâmetros como tensão e iluminação.
- Implementar em sistemas como Linux para otimizar o gerenciamento de tarefas.

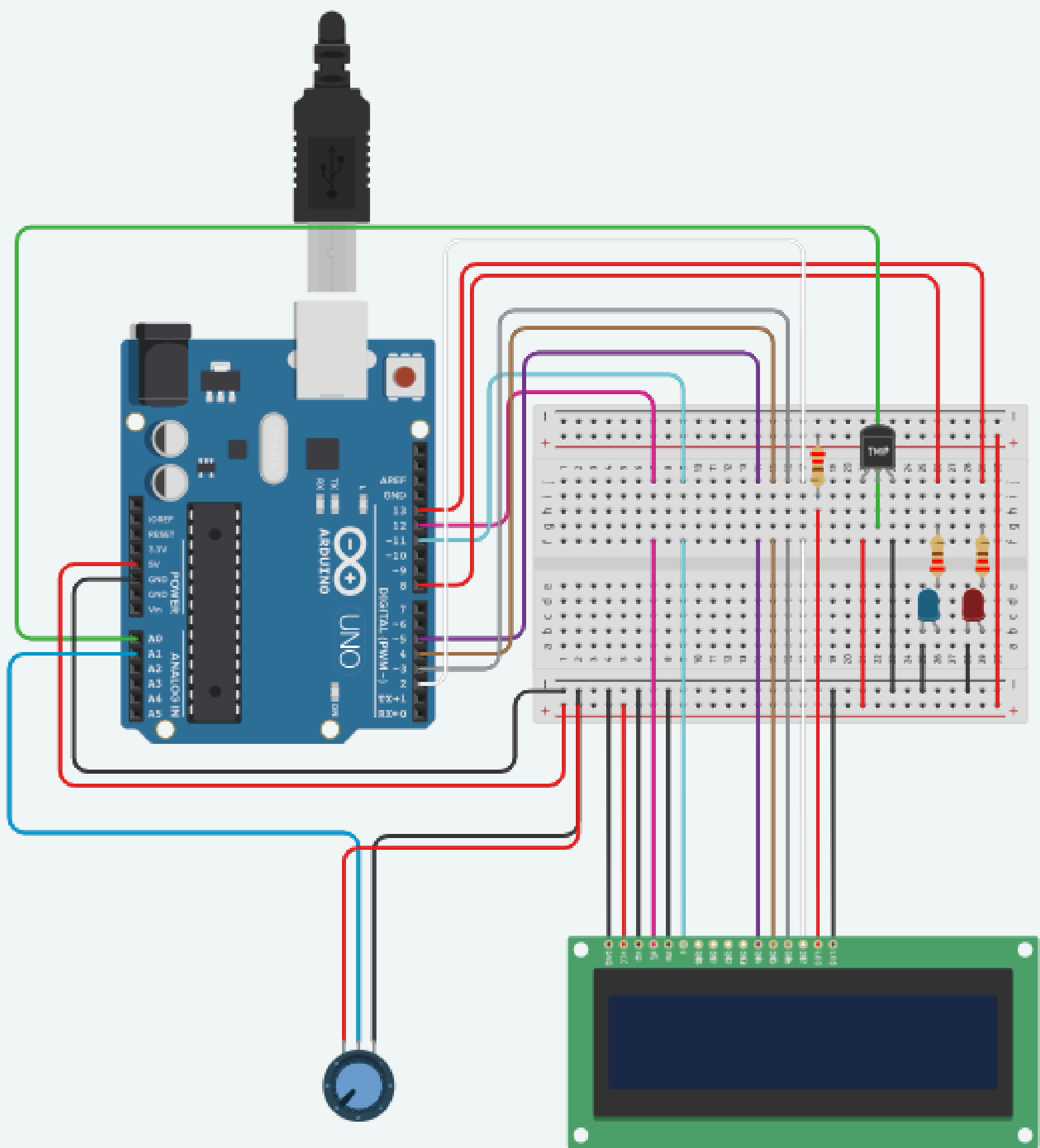
# PROGRAMA USADO



# MICROCONTROLADOR USADO









**CODIGO**



```
1 // Carrega a Biblioteca LiquidCrystal
2 #include <LiquidCrystal.h>
3
4 // Define os pinos que serão utilizados para ligação ao display
5 LiquidCrystal LCD(12,11,5,4,3,2);
6
7 // Variáveis
8
9 int SensorTempPino=0;
10 int SensorUmidPino=1;
11 int porcem=0;
12
13
14 // Define o pino 8 para o alerta de temperatura baixa
15 int AlertaTempBaixa=8;
16 // Define o pino 13 para o alerta de temperatura alta
17 int AlertaTempAlta=13;
18
19 // Define temperatura baixa como abaixo de zero grau Celsius
20 int TempBaixa=0;
21 // Define temperatura alta como acima de 40 graus Celsius
22 int TempAlta=40;
23
```

```
1 void setup()
2 {
3     // Informa se os pinos dos LEDs são de entrada ou saída
4     pinMode(AlertaTempBaixa, OUTPUT);
5     pinMode(AlertaTempAlta, OUTPUT);
6
7     // Define LCD 16 colunas por 2 linhas
8     LCD.begin(16,2);
9
10    //Posiciona o cursor na coluna 0, linha 0;
11    LCD.setCursor(0,0);
12
13    // Imprime a mensagem no LCD
14    LCD.print("Temper.:");
15
16    // Imprime a mensagem no LCD
17    LCD.print("Umidad.:");
18
19    // Muda o cursor para a primeira coluna e segunda linha do LCD
20    LCD.setCursor(0,1);
21
22    // Imprime a mensagem no LCD
23    LCD.print("      C      ");
24
25    // Imprime a mensagem no LCD
26    LCD.setCursor(14,1);
27    LCD.print("%");
28 }
```

```
1 void loop()
2 {
3     // Faz a leitura da tensao no Sensor de Temperatura
4     int SensorTempTensao=analogRead(SensorTempPino);
5
6     // Converte a tensao lida
7     float Tensao=SensorTempTensao*5;
8     Tensao/=1024;
9
10    // Converte a tensao lida em Graus Celsius
11    float TemperaturaC=(Tensao-0.5)*100;
12
13    // Muda o cursor para a primeira coluna e segunda linha do LCD
14    LCD.setCursor(0,1);
15
16    // Imprime a temperatura em Graus Celsius
17    LCD.print(TemperaturaC);
18
19
20    // Faz a leitura da tensao no Sensor de Umidade
21    int SensorUmidTensao=analogRead(SensorUmidPino);
22
23    // Converte a tensao lida em porcentagem
24    float porcem=map(SensorUmidTensao,0,1023,0,100);
25
26    // Muda o cursor para a segunda coluna e nona linha do LCD
27    LCD.setCursor(8,1);
28
29    // Imprime a umidade em porcentagem
30    LCD.print(porcem);
31
32
33    // Acende ou apaga os alertas luminosos de temperatura baixa e alta
34    if (TemperaturaC>=TempAlta) {
35        digitalWrite(AlertaTempBaixa, LOW);
36        digitalWrite(AlertaTempAlta, HIGH);
37    }
38    else if (TemperaturaC<=TempBaixa) {
39        digitalWrite(AlertaTempBaixa, HIGH);
40        digitalWrite(AlertaTempAlta, LOW);
41    }
42    else {
43        digitalWrite(AlertaTempBaixa, LOW);
44        digitalWrite(AlertaTempAlta, LOW);
45    }
46
47    // Aguarda 1 segundo
48    delay(1000);
49 }
```

