



UFRJ

UNIVERSIDADE FEDERAL
DO RIO DE JANEIRO

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

UFRJ Vert-IoT User Guide

João Victor Gioseffi Maciel Alegrio Lopes - 118092648

Rio de Janeiro
Outubro de 2024

Conteúdo

1	Introdução	2
2	Implementação	3
2.1	Materiais	3
2.2	Código	3
2.3	Circuito	6
3	Conclusão	8
3.1	Sugestões para futuras implementações	8

Capítulo 1

Introdução

Para fins de aprendizado, foi estudado e implementado utilizando um microcontrolador arduino, um sistema de monitoramento de clima e solo para irrigação. Neste relatório, irei explicar como replicar o sistema.

Capítulo 2

Implementação

A turma foi dividida em grupos para trabalharem separadamente em cada componente. Os grupos se dividiram entre o sensor de umidade do solo mais visualização dos dados, Mini bomba de água e sensor de chuva mais sensor de umidade e temperatura.

Após todos os grupos entenderem o funcionamento de seus componentes, juntamos eles em um unico circuito e código. Ao longo desse capítulo será explicado o necessário para replicar o sistema.

2.1 Materiais

Para a realização do sistema, será necessário os seguintes materiais:

1x Arduino Uno (micro controlador)

1x soil moisture sensor (sensor de umidade do solo)

1x DHT11 Sensor Temperature and humidity (Sensor de temperatura e umidade)

1x Rain Sensor (Sensor de chuva)

1x Mini Bomba de água

1x fonte de alimentação 5v

1x MOSFET

1x Protoboard

Além disso, um computador com Arduino IDE e alguns cabos para a ligação dos componentes.

2.2 Código

O código a seguir implementa todo o sistema e cria um site onde é possível visualizar todas as variáveis. Ele também define quais valores devem ser considerados para o solo estar molhado ou seco. Além de estar totalmente comentado para melhor entendimento.

```
\diagram{#include "WiFiS3.h"

// Definições para o sensor de umidade do solo
#define soilWet 500 // Valor analógico considerado 'solo molhado'
#define soilDry 750 // Valor analógico considerado 'solo seco'
#define sensorPower 7 // Pino de alimentação do sensor
#define sensorPin A0 // Pino analógico onde o sensor está conectado

// WiFi
char ssid[] = "wPEM"; // your network SSID (name)
char pass[] = "ufrjmecanica"; // your network password (use for WPA, or use as
key for WEP)
```

```

int status = WL_IDLE_STATUS;
WiFiServer server(80);

// Variáveis de exemplo para outros sensores
bool rainDetected = false; // Simula sensor de chuva (substitua por leitura
    real, se disponível)
float temperature = 26.5; // Simula leitura de temperatura
int humidity = 60; // Simula leitura de umidade do ar
int soilMoisture = 0; // Variável para armazenar a leitura do sensor de
    umidade do solo

void setup() {
    Serial.begin(9600);
    pinMode(sensorPower, OUTPUT);
    digitalWrite(sensorPower, LOW); // Inicialmente mantenha o sensor desligado

    // Tentativa de conexão WiFi
    while (status != WL_CONNECTED) {
        Serial.print("Tentando conectar a rede: ");
        Serial.println(ssid);
        status = WiFi.begin(ssid, pass);
        delay(10000);
    }

    server.begin(); // Inicia o servidor
    printWifiStatus(); // Mostra o status da conexão
}

void loop() {
    // Atualiza o valor da umidade do solo
    int moisture = readSoilMoisture();
    Serial.print("Umidade do solo: ");
    Serial.println(moisture);
    // Determine status of our soil
    if (moisture < soilWet) {
        Serial.println("Status: Soil is too wet");
    } else if (moisture >= soilWet && moisture < soilDry) {
        Serial.println("Status: Soil moisture is perfect");
    } else {
        Serial.println("Status: Soil is too dry — time to water!");
    }
    delay(1000); // Take a reading every second for testing

    WiFiClient client = server.available(); // Espera por clientes

    if (client) { // Se houver um cliente conectado
        Serial.println("Novo cliente conectado");
        String currentLine = "";
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();
                Serial.write(c); // Debugging no monitor serial

                if (c == '\n' && currentLine.length() == 0) {
                    // Envia a resposta HTTP (headers e conteúdo HTML)
                    client.println("HTTP/1.1 200 OK");
                    client.println("Content-type:text/html");
                    client.println();

                    // Verifica se a requisição foi para "/sensor-data" e retorna JSON

```

```

if (currentLine.endsWith("GET /sensor-data")) {
    // Envia dados dos sensores no formato JSON
    client.println("{");
    client.print("\"rain\":");
    client.print(rainDetected ? "true" : "false");
    client.println(",");
    client.print("\"soilMoisture\":");
    client.print(moisture); // Envia o valor real de umidade do solo
    client.println(",");
    client.print("\"temperature\":");
    client.print(temperature);
    client.println(",");
    client.print("\"humidity\":");
    client.print(humidity);
    client.println("}");
} else {
    // Caso contrário, retorna a página HTML com os sensores
    client.println("<!DOCTYPE html><html lang='en'><head>");
    client.println("<meta charset='UTF-8'><meta name='viewport' content="
        + "width=device-width, initial-scale=1.0'>");
    client.println("<title>Weather Station Dashboard</title>");
    client.println("<style>body{font-family: Arial, sans-serif; background-
        color:#f4f4f9;text-align:center;padding:20px;}");
    client.println("h1{color:#333;}.sensor-data{display:flex;justify-
        content:space-around;flex-wrap:wrap;margin-top:20px;}");
    client.println(".sensor{background-color:#fff;border-radius:10px;box-
        shadow:0 0 10px rgba(0,0,0,0.1);width:200px;padding:20px;margin
        :10px;}");
    client.println(".sensor h2{font-size:1.2em;color:#444;}.sensor p{
        font-size:1.5em;color:#000;}</style></head>");
    client.println("<body><h1>Weather Station Data</h1><div class='
        sensor-data'>");
    client.println("<div class='sensor'><h2>Rain Sensor</h2><p id='rain
        '>--</p></div>");
    client.println("<div class='sensor'><h2>Soil Moisture</h2><p id='
        moisture'>" + String(moisture) + "</p></div>");
    client.println("<div class='sensor'><h2>Temperature</h2><p id='
        temperature'>--</p></div>");
    client.println("<div class='sensor'><h2>Humidity</h2><p id='humidity
        '>--</p></div></div>");

    // Script para atualizar os dados em tempo real usando AJAX
    client.println("<script>function updateSensorData() {");
    client.println("fetch('/sensor-data').then(response=>response.json()
        ).then(data=>{");
    client.println("document.getElementById('rain').innerText=data.rain
        ? 'Rain Detected': 'No Rain';");
    client.println("document.getElementById('moisture').innerText=data.
        soilMoisture+'%';");
    client.println("document.getElementById('temperature').innerText=
        data.temperature+'°C';");
    client.println("document.getElementById('humidity').innerText=data.
        humidity+'%';");
    client.println("}).catch(error=>console.error('Error fetching sensor
        data:', error));}");

    // Atualiza os dados a cada 5 segundos
    client.println("setInterval(updateSensorData,5000);");
    client.println("updateSensorData();");
    client.println("</script></body></html>");

```

```

        }
        break;
    }
    if (c == '\n') {
        currentLine = "";
    } else if (c != '\r') {
        currentLine += c;
    }
}
}
client.stop(); // Fecha a conexão com o cliente
Serial.println("Cliente desconectado");
}
}

// Função para ler a umidade do solo e mapear para uma escala de 0 a 100%
int readSoilMoisture() {
    digitalWrite(sensorPower, HIGH); // Liga o sensor
    delay(10); // Aguarda o sensor estabilizar
    int val = analogRead(sensorPin); // Lê o valor analógico do sensor
    digitalWrite(sensorPower, LOW); // Desliga o sensor
    return val; // Retorna o valor lido
}

void printWifiStatus() {
    // Exibe o SSID da rede conectada
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    // Exibe o endereço IP da placa
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);

    // Exibe a força do sinal
    long rssi = WiFi.RSSI();
    Serial.print("signal strength (RSSI):");
    Serial.print(rssi);
    Serial.println(" dBm");

    // Exibe o endereço para acessar no navegador
    Serial.print("Para acessar a página, abra o navegador em http://");
    Serial.println(ip);
}
}
}

```

2.3 Circuito

A Figura 2.1 mostra o esquema do circuito montado.

Legenda:

- 1 - Arduino Uno
- 2- Sensor de Umidade do Solo
- 3- Sensor de Chuva
- 4- Sensor de umidade e temperatura
- 5- Fonte de 5v
- 6- MOSFET
- 7- Mini Bomba de água

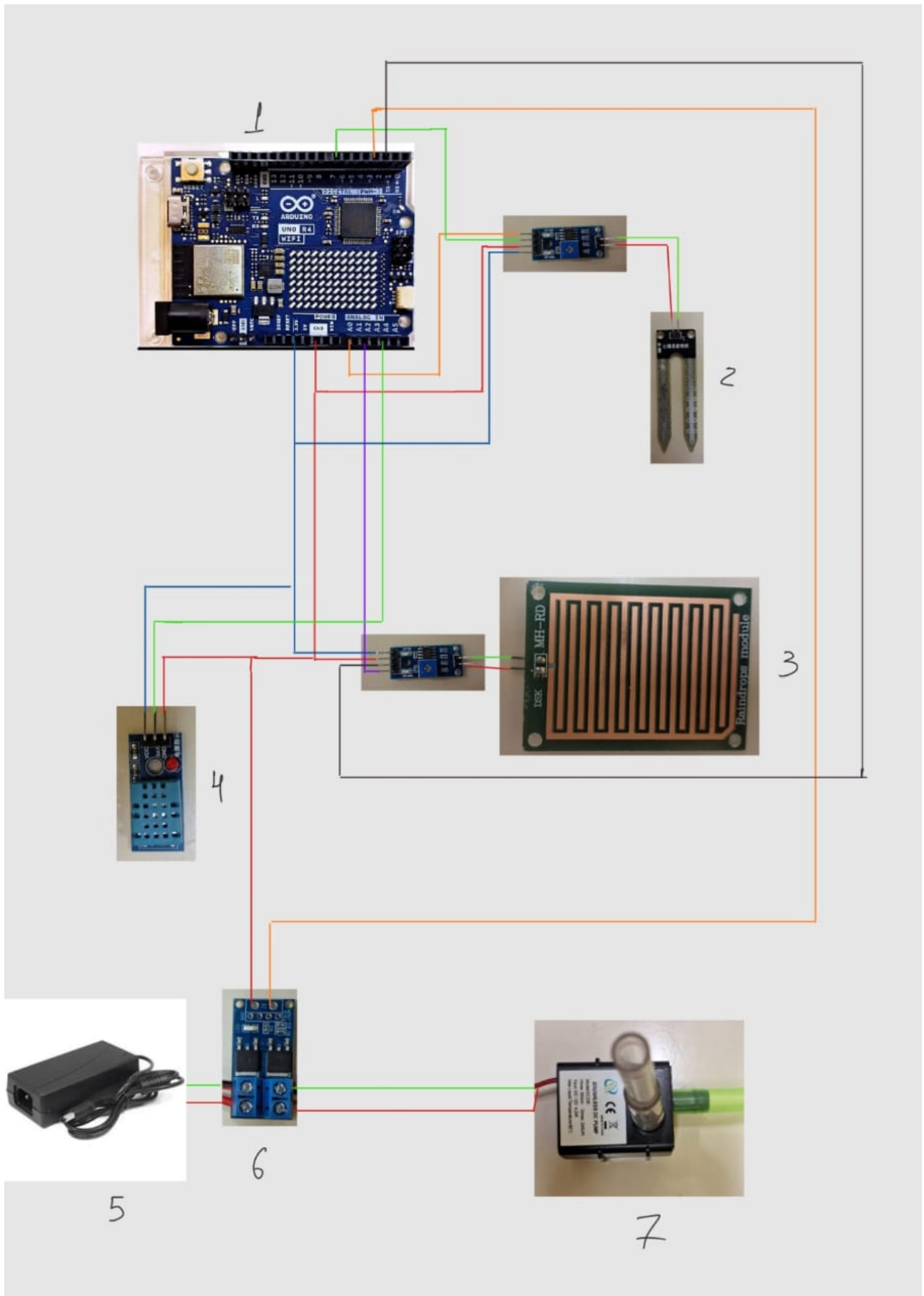


Figura 2.1: Enter Caption

Capítulo 3

Conclusão

O sistema monitora as seguintes variáveis: Umidade do Solo, Umidade do Ar, Temperatura e Presença de chuva. Além disso, o MOSFET ativa a Mini Bomba de Água quando o sensor de umidade do solo apresenta o solo como Seco.

3.1 Sugestões para futuras implementações

No presente momento, os sensores de umidade, temperatura e chuva servem somente para medição dessas mesmas variáveis. Para implementações futuras, a implementação de diminuir a sensibilidade para um solo seco em altas temperaturas, pois um solo muito irrigado em altas temperaturas pode fazer um efeito similar de cozimento à planta.

Além disso, a bomba não ligar em eminência de chuva, a fim de poupar a água do reservatório da mini bomba de água.

Repositório GitHub

O código está disponível no seguinte link: [college/Transcal II/Sensores.ino at main · alegrio/college](#) para acesso mais fácil.

Bibliografia

- [1] ASWINTH. *Soil moisture sensor with Arduino*. Arduino Project Hub, 2018. Disponível em: <https://projecthub.arduino.cc/Aswinth/soil-moisture-sensor-with-arduino-91c818>. Acesso em: 27 out. 2024.
- [2] EDWARDTHE. *How to use raindrop sensor with Arduino Leonardo*. Arduino Project Hub, 2022. Disponível em: <https://projecthub.arduino.cc/edwardthe/how-to-use-raindrop-sensor-with-arduino-leonardo-acf24d>. Acesso em: 27 out. 2024.
- [3] RUDRAKSH2008. *Temperature and humidity sensor with Arduino*. Arduino Project Hub, 2021. Disponível em: <https://projecthub.arduino.cc/rudraksh2008/temperature-and-humidity-sensor-with-arduino-1d52a6>. Acesso em: 27 out. 2024.
- [4] Arduino. *WiFi examples for Arduino UNO R4 WiFi*. Arduino Documentation, 2024. Disponível em: <https://docs.arduino.cc/tutorials/uno-r4-wifi/wifi-examples/>. Acesso em: 27 out. 2024.