**UC: Internet das Coisas IOT**

Relatório de Software: Horta Inteligente

**Integrantes:**

* Carlos Aguiar – Pesquisador de Hardware
* Gustavo Rodrigues – Programador
* Miguel Mernick – Responsável pela simulação
* Stefany Lino – Responsável por testes

**CONTEXTO**

Uma escola técnica deseja modernizar sua horta experimental, utilizada em aulas práticas, por meio da automação e monitoramento inteligente de variáveis como umidade do solo, temperatura, luminosidade e irrigação. A proposta é que os alunos do curso de Desenvolvimento de Sistemas, em grupos, desenvolvam uma solução integrada de software e hardware capaz de coletar dados dos sensores, armazenar informações e controlar automaticamente os dispositivos, como a bomba d’água, com base nas condições do ambiente.

**PESQUISA**

Para a montagem do sistema de irrigação automatizado com IoT, foram utilizados diversos componentes selecionados por sua funcionalidade e viabilidade técnica. Optou-se por uma placa Arduino UNO devido ao seu baixo custo e à estabilidade durante os testes, especialmente após falhas no simulador do ESP32. Um módulo Wi-Fi foi incorporado para permitir conectividade remota com a plataforma IoT. A irrigação é controlada por uma válvula solenoide 12V (rosca ½”), escolhida em vez de uma minibomba, pois se conecta diretamente à rede de água pressurizada e fornece vazão suficiente para a horta. Um sensor de umidade do solo foi usado para medir a necessidade de irrigação com base na resistência elétrica do solo, enquanto sensores de luminosidade e temperatura monitoram as condições ambientais. O acionamento da válvula é feito por um módulo relé de 5V, que recebe sinal do Arduino. Todo o sistema é alimentado por uma fonte de 12V, e uma mangueira é utilizada para canalizar a água até a área de irrigação. Esses componentes juntos permitem a automação eficiente e de baixo custo do sistema de irrigação.

**PROGRAMAÇÃO**

Este código Arduino implementa um sistema de monitoramento ambiental utilizando sensores de umidade do solo, temperatura e luminosidade, com controle de um servo motor com base em limiares definidos. O sistema lê continuamente os dados dos sensores e, caso alguma das condições críticas seja atingida — solo seco, temperatura elevada ou alta intensidade de luz —, o servo é ativado, realizando movimentos de 0° a 180°, simulando a ação de um atuador, como uma válvula ou irrigador. Caso contrário, o servo permanece em repouso na posição central (90°). As leituras são exibidas no monitor serial para acompanhamento em tempo real, e a lógica do sistema é simples, sendo adequada para protótipos de irrigação automatizada ou sistemas de resposta ambiental básica.

#include <Servo.h> // Inclui a biblioteca para controlar servo motores

Servo servo; // Cria um objeto chamado 'servo' para usar depois

// Define os pinos onde os sensores estão conectados

const int pinUmidade = A0; // Sensor de umidade no pino A0

const int pinTemperatura = A1; // Sensor de temperatura no pino A1

const int pinLuz = A2; // Sensor de luz no pino A2

const int pinServo = 9; // Servo motor conectado no pino digital 9

// Define os valores que serão usados para tomar decisões

const int limiarUmidade = 400; // Valor mínimo aceitável de umidade

const float limiarTemperatura = 30; // Temperatura máxima permitida (em graus Celsius)

const int limiarLuz = 500; // Valor de luz acima do qual será considerada "alta"

void setup() {

Serial.begin(9600); // Inicia a comunicação com o computador (Serial Monitor)

servo.attach(pinServo); // Diz ao Arduino em qual pino o servo está conectado

servo.write(90); // Coloca o servo na posição inicial (meio)

}

void loop() {

// ======= LEITURA DO SENSOR DE TEMPERATURA =======

int valorTemp = analogRead(pinTemperatura); // Lê o valor do sensor (0 a 1023)

float tensao = valorTemp \* (5.0 / 1023.0); // Converte o valor lido em tensão (0 a 5V)

float temperaturaC = (tensao - 0.5) \* 100.0; // Converte a tensão para temperatura em °C

// ======= LEITURA DO SENSOR DE UMIDADE =======

int valorUmidade = analogRead(pinUmidade); // Lê o valor do sensor de umidade

// ======= LEITURA DO SENSOR DE LUZ =======

int valorLuz = analogRead(pinLuz); // Lê o valor do sensor de luz

// ======= MOSTRA OS VALORES NO MONITOR SERIAL =======

Serial.print("Umidade: ");

Serial.print(valorUmidade); // Mostra a umidade lida

Serial.print(" | Temperatura: ");

Serial.print(temperaturaC); // Mostra a temperatura em °C

Serial.print(" | Luz: ");

Serial.println(valorLuz); // Mostra a intensidade de luz

// ======= DECISÃO: DEVE ATIVAR O SERVO? =======

if (valorUmidade > limiarUmidade || // Se a umidade for baixa

temperaturaC > limiarTemperatura || // OU a temperatura for alta

valorLuz > limiarLuz) { // OU a luz estiver forte

Serial.println(">>> Alguma condição atendida - Servo girando!");

servo.write(0); // Gira o servo para 0° (um lado)

delay(500); // Espera 0,5 segundos

servo.write(180); // Gira o servo para 180° (outro lado)

delay(500); // Espera mais 0,5 segundos

} else {

Serial.println(">>> Nenhuma condição - Servo parado.");

servo.write(90); // Servo na posição central (parado)

}

Serial.println("--------------------------"); // Linha separadora no Serial Monitor

delay(500); // Espera meio segundo antes de repetir tudo

}

**CIRCUITO**

**TESTES**

Para iniciar os testes, os requisitos foram revisitados