**1. Introdução**

Este projeto implementa um sistema de monitoramento com um ESP32, utilizando múltiplos sensores para capturar dados ambientais. Os sensores utilizados são:

* **DHT22**: para medir temperatura e umidade.
* **HC-SR04**: para medir distância, utilizado para detecção de nível de água.
* **PIR**: para detectar movimento.
* **LDR**: para medir a luminosidade ambiente.

Os dados coletados são exibidos no monitor serial, oferecendo uma visão contínua dos valores monitorados.

**2. Bibliotecas**

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <DHT\_U.h>

Estas bibliotecas são necessárias para o funcionamento do sensor DHT22. A Adafruit\_Sensor fornece uma interface unificada para sensores, e as bibliotecas DHT e DHT\_U são específicas para sensores de temperatura e umidade DHT.

**3. Definição de Pinos**

#define DHTPIN 15 // Pino do DHT22

#define DHTTYPE DHT22 // Tipo de sensor DHT

#define TRIG\_PIN 12 // Pino TRIG do HC-SR04

#define ECHO\_PIN 14 // Pino ECHO do HC-SR04

#define PIR\_PIN 13 // Pino do sensor PIR

#define LDR\_PIN 34 // Pino analógico do LDR

Aqui definimos os pinos que o ESP32 utilizará para se comunicar com cada sensor. Esses pinos são configurados de acordo com o esquema de hardware

**4. Inicialização do Sensor DHT22**

DHT dht(DHTPIN, DHTTYPE);

Este comando cria uma instância do sensor DHT22, permitindo que o ESP32 leia os valores de temperatura e umidade.

**5. Função setup()**

void setup() {

Serial.begin(115200);

dht.begin();

pinMode(TRIG\_PIN, OUTPUT);

pinMode(ECHO\_PIN, INPUT);

pinMode(PIR\_PIN, INPUT);

}

A função setup() é executada uma vez, ao iniciar o programa:

* Serial.begin(115200): Configura a taxa de transmissão serial para 115200 bps.
* dht.begin(): Inicializa o sensor DHT22.
* pinMode(): Define os modos dos pinos:
  + TRIG\_PIN: Saída para enviar o pulso de disparo do HC-SR04.
  + ECHO\_PIN: Entrada para receber o pulso de retorno do HC-SR04.
  + PIR\_PIN: Entrada para ler o estado do sensor PIR.

**6. Função loop()**

A função loop() é executada repetidamente para ler os sensores e exibir os resultados.

**Leitura do DHT22**

float humidity = dht.readHumidity();

float temperature = dht.readTemperature();

if (isnan(humidity) || isnan(temperature)) {

Serial.println("Erro ao ler o DHT22!");

} else {

Serial.print("Umidade: ");

Serial.print(humidity);

Serial.print(" %\t");

Serial.print("Temperatura: ");

Serial.print(temperature);

Serial.println(" \*C");

}

* Lê a umidade e temperatura.
* Verifica se a leitura é válida (isnan()).
* Exibe os valores no monitor serial.

**Leitura do HC-SR04**

digitalWrite(TRIG\_PIN, LOW);

delayMicroseconds(2);

digitalWrite(TRIG\_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(TRIG\_PIN, LOW);

long duration = pulseIn(ECHO\_PIN, HIGH);

float distance = duration \* 0.034 / 2;

if (distance > 0) {

Serial.print("Distância: ");

Serial.print(distance);

Serial.println(" cm");

} else {

Serial.println("Erro ao ler o HC-SR04!");

}

* Envia um pulso de 10µs para o TRIG.
* Mede a duração do pulso retornado no ECHO.
* Calcula a distância com base na fórmula: distância = (duração \* 0.034) / 2.
* Exibe a distância no monitor serial.

**Leitura do PIR**

int pirState = digitalRead(PIR\_PIN);

if (pirState == HIGH) {

Serial.println("Movimento detectado!");

} else {

Serial.println("Sem movimento.");

}

* Lê o estado do sensor PIR.
* Exibe no monitor serial se movimento foi detectado.

**Leitura do LDR**

int ldrValue = analogRead(LDR\_PIN);

Serial.print("Luminosidade (LDR): ");

Serial.println(ldrValue);

* Lê o valor analógico do LDR (entre 0 e 4095).
* Exibe o nível de luminosidade no monitor serial.

**Resumo dos Sensores Utilizados:**

1. **DHT22 (Sensor de Temperatura e Umidade):**
   * Mede a temperatura e a umidade do ambiente.
   * É usado em projetos de monitoramento ambiental e controle de climatização.
2. **HC-SR04 (Sensor Ultrassônico):**
   * Mede a distância até um obstáculo usando ondas ultrassônicas.
   * Usado para medir o nível de líquidos ou detectar a proximidade de objetos.
3. **PIR (Passive Infrared Sensor):**
   * Detecta movimento baseado em mudanças na radiação infravermelha.
   * Comumente utilizado em sistemas de alarme e automação residencial.
4. **LDR (Light Dependent Resistor):**
   * Mede a intensidade da luz ambiente.
   * Usado para controle de iluminação e monitoramento de luminosidade.