# Documentação do Projeto: Monitoramento de Nutrientes e Nível de Água com ESP32

## 1. Introdução

Este projeto utiliza um microcontrolador ESP32 para monitorar a presença de Potássio (K) e Fósforo (P) no solo, além de medir a umidade e a luminosidade. O sistema utiliza dois botões para alternar o estado de presença de nutrientes e um sensor DHT22 para medir a umidade e a temperatura. O nível de luminosidade é medido por um LDR. Os dados são exibidos no monitor serial em formato JSON sempre que o nível de umidade for considerado baixo.

## 2. Componentes Utilizados

- ESP32  
- 2 Botões  
- DHT22 (sensor de umidade e temperatura)  
- LDR (Light Dependent Resistor)  
- LED  
- Resistor de 10 kΩ para o LDR  
- Resistor de 220 Ω para o LED

## 3. Conexões

As conexões dos componentes com o ESP32 são as seguintes:  
- Botão K (Potássio): GPIO 14, conectado ao GND.  
- Botão P (Fósforo): GPIO 27, conectado ao GND.  
- DHT22: VCC ao 3.3V, DATA ao GPIO 4, GND ao GND, resistor de 10 kΩ entre DATA e VCC.  
- LDR: Um terminal ao 3.3V, outro terminal ao GPIO 34 e ao GND com resistor de 10 kΩ.  
- LED: Anodo ao GPIO 18, catodo ao GND com resistor de 220 Ω.

## 4. Funcionamento do Código

O código realiza as seguintes operações:  
- Inicializa os sensores e configura os pinos dos botões e do LED.  
- Lê o estado dos botões e alterna o estado de presença de Potássio (K) e Fósforo (P).  
- Lê a umidade e a temperatura usando o sensor DHT22.  
- Lê o valor da luminosidade usando o LDR e converte para uma porcentagem.  
- Acende o LED se a umidade estiver abaixo de 30%, indicando um nível de água baixo.  
- Quando o LED acende, os dados são exibidos no monitor serial em formato JSON.

## 5. Estrutura do JSON

Os dados exibidos no monitor serial seguem o seguinte formato JSON:

{  
 "humidity": <valor da umidade>,  
 "light\_percentage": <percentual de luminosidade>,  
 "potassium\_present": <true/false>,  
 "phosphorus\_present": <true/false>  
}

## 6. Explicação das Funções

### 6.1. setup()

Inicializa os sensores e configura os pinos dos botões e do LED. Também inicia o monitor serial para exibir os dados.

### 6.2. loop()

Verifica o estado dos botões e alterna o estado dos nutrientes. Lê os valores de umidade, temperatura e luminosidade. Acende o LED se o nível de umidade estiver baixo e exibe os dados em JSON no monitor serial.

### 6.3. checkButton()

Verifica o estado dos botões utilizando debounce para evitar leituras incorretas. Alterna o estado do nutriente correspondente ao botão pressionado.

### 6.4. readLightLevel()

Lê o valor analógico do LDR e converte para uma porcentagem de luminosidade (0% a 100%).

### 6.5. printJson()

Cria um documento JSON com os valores de umidade, luminosidade, presença de Potássio (K) e Fósforo (P), e exibe o JSON formatado no monitor serial.

## 7. Exemplo de Saída no Monitor Serial

Dados em formato JSON:  
{  
 "humidity": 25.4,  
 "light\_percentage": 37.0,  
 "potassium\_present": true,  
 "phosphorus\_present": false  
}

## 8. Conclusão

Este projeto demonstra como utilizar o ESP32 para monitorar condições de umidade e luminosidade, além de controlar o estado de nutrientes no solo usando botões. Os dados são exibidos em formato JSON no monitor serial, facilitando a visualização e integração com outros sistemas.