**Manual de Instalação e Configuração do Sistema de Monitoramento de Gás**

**Versão: 1.1 | Data: Dezembro de 2024**

**1. Introdução**

**Este manual detalha o processo completo de instalação e configuração do sensor de gás MQ-2 integrado a um sistema de monitoramento que inclui coleta de dados via API, armazenamento em banco de dados e exibição em um dashboard web.**

**2. Requisitos**

**Equipamentos Necessários**

**Hardware:**

**1 - Sensor de Gás MQ-2**

**2 - Placa Arduino Uno R3**

**3 - Fonte de alimentação para Arduino (USB conectado ao Computador)**

**4 - Protoboard**

**5 - Fios jumper**

**6 – Notebook**

**3. Software:**

**1 - Arduino IDE (última versão)**

**2 - Node.js (para a API)**

**3 – MySQL**

**4 – Virtual Machine**

**5 – Visual Code (HTML, CSS, Javascript, API)**

**6 – GitHub (Versionamento do Projeto)**

**Ferramentas Necessárias (ferramentas utilizadas para implantar o sensor)**

**4. Requisitos do Ambiente:**

**1 - Rede Wi-Fi estável para integração com API e dashboard**

**2 - Computador ou servidor próximo ao local de instalação para processamento de dados**

**3 – Entrada de gás livre para implantação do Arduino R3 (MQ2, Protoboard)**

**5. Procedimento de Instalação**

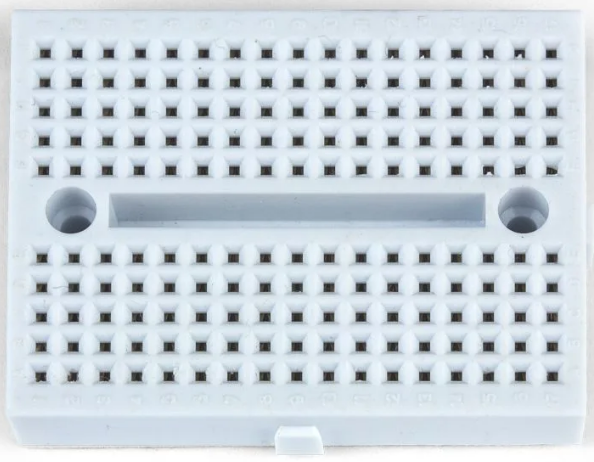
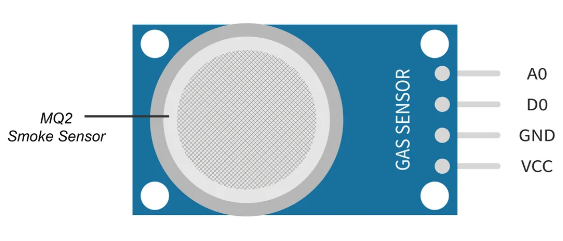
**5.1. Preparação do Local**

**Escolha um local elevado e limpo para instalar o sensor.**

**Imagem de um exemplo de cozinha para a instalação**

**Garanta fácil acesso ao sensor para manutenção.**

**6. Conexões com o Sensor MQ-2**

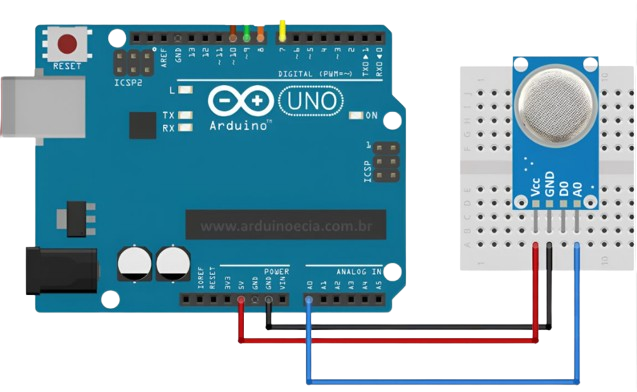
**1. Conecte o sensor à protoboard.**

**2. Faça as conexões dos fios (Jumpers):**

**VCC do sensor ao 5V do Arduino. (Jumper Red)**

**GND do sensor ao GND do Arduino. (Jumper Black)**

**(A OUT) ao A0 do Arduino. (Jumper Blue)**

**Verifique que todas as conexões estejam corretas e firmes.**

**7. Configuração do Código no Arduino**

**1. Instale a IDE (Arduino IDE)**

**2. Inicie a IDE**

**3. Escreva o Código e compile para verificar**

**8. Conexão do Arduino via USB a o computador para a transferência do código**

**1. Conecte o arduino montado ao computador via USB**

**2. Transfira o código para o arduino conectado (Certifique que o código tenha sido executado antes de transferir, para evitar problemas no código)**

**3. Verificar se o led do sensor está ligado (led red), após a transferência do código.**

**2. Crie a API para enviar dados do sensor ao banco:**

**from flask import Flask, request**

**import mysql.connector**

**app = Flask(\_\_name\_\_)**

**db = mysql.connector.connect(**

**host="localhost",**

**user="root",**

**password="password",**

**database="gas\_monitoring"**

**)**

**@app.route('/api/data', methods=['POST'])**

**def save\_data():**

**data = request.json**

**cursor = db.cursor()**

**cursor.execute("INSERT INTO readings (sensor\_value) VALUES (%s)", (data['value'],))**

**db.commit()**

**return {"status": "success"}**

**if \_\_name\_\_ == "\_\_main\_\_":**

**app.run(host="0.0.0.0", port=5000)**

**3. Teste a API enviando dados fictícios:**

**curl -X POST -H "Content-Type: application/json" -d '{"value": 300}' http://localhost:5000/api/data**

**3.5. Configuração do Banco de Dados**

**1. Instale o MySQL e configure o banco:**

**CREATE DATABASE gas\_monitoring;**

**CREATE TABLE readings (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**sensor\_value INT NOT NULL,**

**timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP**

**);**

**2. Teste a conexão da API com o banco de dados.**

**3.6. Configuração do Dashboard**

**1. Use um framework web como Flask ou Node.js para criar o dashboard.**

**2. Exemplo básico em Flask para visualizar dados:**

**@app.route('/dashboard', methods=['GET'])**

**def dashboard():**

**cursor = db.cursor()**

**cursor.execute("SELECT \* FROM readings ORDER BY timestamp DESC LIMIT 10")**

**data = cursor.fetchall()**

**return render\_template("dashboard.html", data=data)**

**3. HTML para o dashboard:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Monitoramento de Gás</title>**

**</head>**

**<body>**

**<h1>Últimas Leituras</h1>**

**<table border="1">**

**<tr>**

**<th>ID</th>**

**<th>Valor</th>**

**<th>Timestamp</th>**

**</tr>**

**{% for row in data %}**

**<tr>**

**<td>{{ row[0] }}</td>**

**<td>{{ row[1] }}</td>**

**<td>{{ row[2] }}</td>**

**</tr>**

**{% endfor %}**

**</table>**

**</body>**

**</html>**

**---**

**4. Testes e Validação**

**Teste do Sistema Local**

**Use o monitor serial para validar o funcionamento do sensor.**

**Teste de API**

**Envie dados do sensor para a API e verifique no banco.**

**Teste do Dashboard**

**Acesse o dashboard e confira se as leituras aparecem corretamente.**

**---**

**5. Solução de Problemas**

**Problema: Não há leitura do sensor.**

**Verifique as conexões do sensor com o Arduino.**

**Teste o sensor com outro código simples.**

**Problema: API não conecta ao banco.**

**Verifique as credenciais e permissões no banco.**

**Confirme que o servidor MySQL está ativo.**

**Problema: Dashboard não exibe dados.**

**Verifique a rota da API e a estrutura do banco.**

**---**

**6. Manutenção e Atualização**

**Limpeza regular do sensor para evitar acumulação de poeira.**

**Atualizações do código e integração com o banco de dados conforme novas versões forem lançadas.**

**---**

**7. Contato para Suporte**

**E-mail: suporte@gasbusters.com**

**Telefone: (11) 99999-9999**