



Sensor de gás: Evitar acidentes domésticos com o uso da IoT

Antonio Mira Gomes Neto, Wallace Rodrigues De Santana

¹Universidade Presbiteriana Mackenzie (UPM)
Rua da Consolação, 930 Consolação, São Paulo - SP, 01302-907 – Brazil

Antonioneto6962@gmail.com

Abstract. *This article describes a kitchen gas leak solution project proposed by this duo, using MQ2 sensors and MQTT internet protocol that will notify users on their cell phones about the leak warning form below.*

Resumo. *Este artigo descreve um projeto de solução de vazamento de gás de cozinha proposto por esta dupla, usando sensores MQ2 e protocolo de internet MQTT que notificará os usuários em seus telefones celulares sobre o formulário de aviso de vazamento abaixo.*

1. Introdução

O projeto visa atender às Frequentes e mortais vazamentos de gás. através de um pequeno projeto envolvendo com boa relação custo-benefício, desenvolveremos um detector de gás que será fácil de usar e muito importante poder resolver os problemas do dia a dia a um ótimo preço. Produtos eletrônicos e de tecnologia. Este projeto visa solucionar o problema de vazamento de gás de cozinha, minimizar o número de acidentes. Para executá-lo, vamos desenvolver um software capaz de detectar gases no ar que excedem os valores considerados como sempre. Assim, quando ultrapassado, será enviado um aviso audiovisual, via Alarmes e LEDs para alertar sobre possíveis vazamentos de gás ou qualquer outro Produtos químicos presentes no local, e os alertas serão emitidos por seu smartphone

2. Materiais e Métodos

2.1 Materiais e métodos utilizados

- **Sensor MQ-2**

Sensor de Fumaça e Gás MQ2 O sensor MQ2 (Figura 1) é capaz de detectar concentrações de diversos gases e fumos inflamáveis no ambiente, como o butano, principal componente do gás de cozinha. O sensor também pode captar outros gases. elementos, como GLP, Metano, Propano, Hidrogênio, Álcool e Gás Natural, captura desses elementos através das concentrações, em partículas por milhão (ppm) desses elementos presentes. Em um ambiente, a legibilidade e a exibição variavam de 200 a 10.000 ppm.



Figura 1. Sensor MQ-2

Fonte: www.curtocircuito.com.br/sensor-de-gas-mq-2.html

- **NodeMcu ESP8266**

Os módulos utilizados para desenvolver o projeto são a placa de desenvolvimento NodeMcu ESP8266 (Figura 2), da qual constam várias outras duas placas, sendo delas, o NodeMCU Devkit 1.0 e o módulo ESP12E. O NodeMcu ESP8266 é utilizado como microcontrolador e agente mestre para estabelecer comunicação via WiFi do circuito com a rede do sistema, através do qual também pode monitorar as leituras dos sensores utilizando o protocolo MQTT, a escolha para este microcontrolador é o devido a sua popularidade no ambiente IoT, sua capacidade de suportar o uso dos protocolos MQTT e TCP/IP e sua praticidade, o NodeMCU é uma plataforma open source da família. Esta placa foi iniciada em 2014 e é bem interessante, pois ao contrário de alguns módulos desta família que necessitam de um conversor USB serial externo para que haja troca de informações entre computador e o módulo, o NodeMCU já vem com um conversor USB serial integrado, assim como citado na publicação sobre a utilização do módulo ESP8266 (Oliveira, Greici 2017) [1]



Figura 2. NodeMcu ESP8266

Fonte: www.adrobotica.com/produto/modulo-wifi-esp8266-nodemcu-v3-lolin-com-chip-ch340g/

- **Protoboard 830 pinos**

Utilizada como matriz de contato, a protoboard (Figura 3) é um componente fabricado em plástico, que é responsável pela conexão e transmissão de sinais elétricos entre os componentes do circuito, através de sua função, na qual Sensor acoplado diretamente, à la carte foi escolhido o Pin 830, por garantir espaço suficiente para o acomodar todos os elementos do projeto.

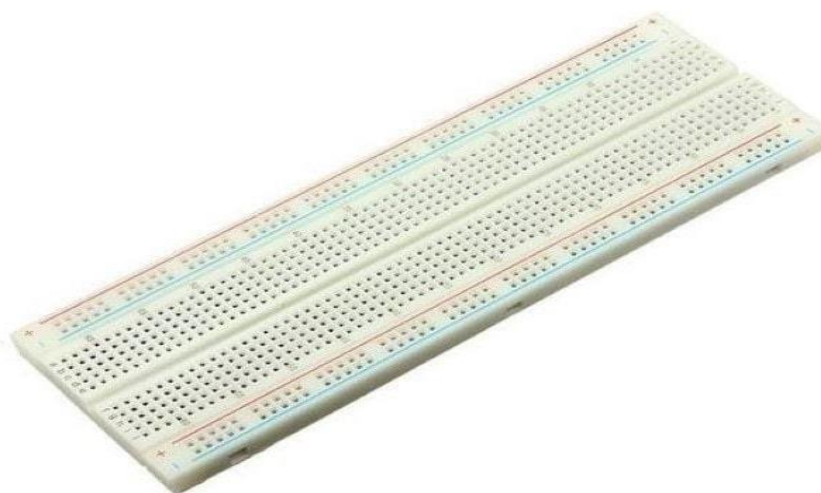


Figura 3. Protoboard 830 pinos

Fonte: www.wjcomponentes.com.br/protoboard-830

- **Buzzer**

Selecionado como atuador do projeto, o Buzzer (Figura 4) tem como alvo primário em um sistema sonoro de notificação de vazamento de gás para a pessoa mais próxima, o emitindo um som semelhante a um alarme de campainha, seu funcionamento primário é devido a o cristal em seu nome, Piezo, que pode ser controlado aplicando mudanças de frequência à tensão que flui pelos condutores da sala, cria assim efeitos sonoros diferentes.



Figura 4. Buzzer Ativo 5V

Fonte: www.piauino.com.br/pd-41c59d-buzzer-5v-continuo-ativo-i1.html

- **Led 5 mm difuso**

Utilizado como indicador visual do projeto, trata-se de um pino de LED comum, que é conectado a todo o circuito via protoboard e responderá aos sinais captados pelo sensor durante o processo. projeto.



Figura 5. LED vermelho

Fonte: www.wjcomponentes.com.br/resistores-leds/leds/led-5-mm-difuso

- **Cabos jumper macho/macho e cabos jumper macho/fêmea**

Responsáveis por conectar uns aos outros, seja através ou não, breadboard, os cabos jumper são fios de cobre com pinos de conexão em suas extremidades, estão disponíveis nos seguintes componentes: macho/macho, macho/fêmea e fêmea/fêmea, no projeto somente foram utilizados cabos macho/macho (Figura 6) e macho/fêmea (Figura 7).

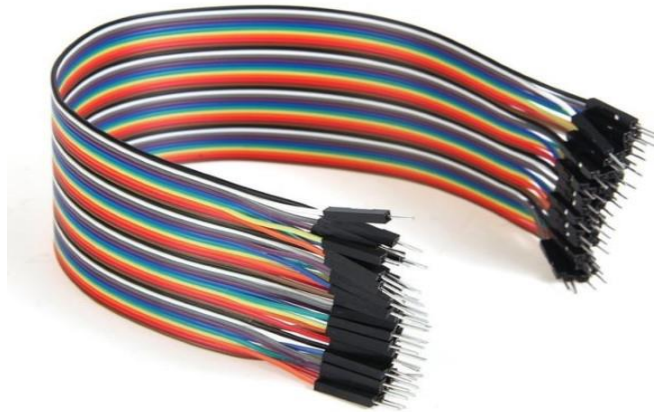


Figura 6. Cabos macho/macho

Fonte: www.filipeflop.com/produto/jumpers-macho-macho-x40-unidades/



Figura 7. Cabos macho/fêmea

Fonte: www.amazon.com.br/Cabo-Jumper-F%C3%A0mea-macho-Protoboard-Arduino/dp/B087WY8ZV5

- **2.2 Protocolos, Interfaces e Módulos de Comunicação**

- **MQTT**

O protocolo de MQTT é responsável por enviar micro mensagens para sensores além de pequenos dispositivos moveis. Este protocolo consiste em uma comunicação entre servidor e cliente, no qual pode tanto enviar, quanto receber insights. O MQTT também utiliza e se baseia no protocolo TCP/IP. Sua utilização é através de métodos chamados, publish e subscribe, os quais enviam informações através de elementos chamados tópicos, os quais são enviados para o Broker Service e tem em sua composição um identificador, no caso um tópico definido, e o conteúdo da sua mensagem, definido como payload, onde esse deve ser do tipo char.

- **Mosquitto MQTT Broker**

Mosquitto MQTT Broker O Mosquitto é um serviço que basicamente recebe todas as mensagens enviadas pelos clientes e direciona aos destinatários. Lembrando que o cliente neste caso pode ser todo e qualquer serviço que possa interagir com o broker, assim como citado na publicação de comunicação com Broker Mosquitto MQTT (Damin, Délio,2019) [2]

- **Arduino IDE**

A IDE de código aberto e utilizada no Arduino está disponível em todas plataformas, entre elas, macOS, Windows e Linux. Suas funções internas são escritas em C ou C++. Através desta IDE, o trabalho é facilitado uma vez que pode-se utilizar qualquer placa Arduino

- **Node-RED**

O Node-RED, é uma ferramenta visual de ambiente de código aberto, que inicialmente foi desenvolvida para implementar, criar e/ou conectar dispositivos de IoT, tendo sido estendida posteriormente para hardwares, APIs e web services. Assim sendo, por meio dos nodes ou nós é possível ler arquivos CSV, escutar eventos http, tcp, websocket, twitter, mqtt entre outros. Ele possui um editor baseado em navegador que além de simples e compatível com todos os browsers, facilita a conexão de fluxos usando os nós (nodes) do palette. E assim como o protocolo MQTT (Message Queue Telemetry Transport), assunto de outras postagens dentro do universo de IoT. (Basílio, Shirley 2021) [3]

• 2.3 Descrição do software desenvolvido

O principal objetivo do software desenvolvido é obter os dados de leitura capturados a partir do sensor de gás e fumaça MQ-2, e ter esses dados em constante verificação, e assim que esses dados ultrapassem um determinado valor estabelecido no código o sistema terá a responsabilidade de comunicar ao usuário enviando sinais para alteração de estado de atuadores e também envio constante dos dados para o telegrama alertando que há um vazamento de gás. O software utilizado foi desenvolvido através da plataforma Arduino IDE, utilizando as bibliotecas ESP8266WiFi e PubSubClient, onde a primeira oferece uma série de classes, métodos e propriedades para o estabelecimento da comunicação através do WiFi utilizando um microcontrolador que contenha um módulo WiFi, ou apenas um módulo WiFi, enquanto a segunda é responsável por prover um cliente para aplicação dos métodos de publish e subscribe, principais métodos utilizados pelo protocolo MQTT, para envio e recebimento de informações durante a comunicação entre dois ou mais dispositivos

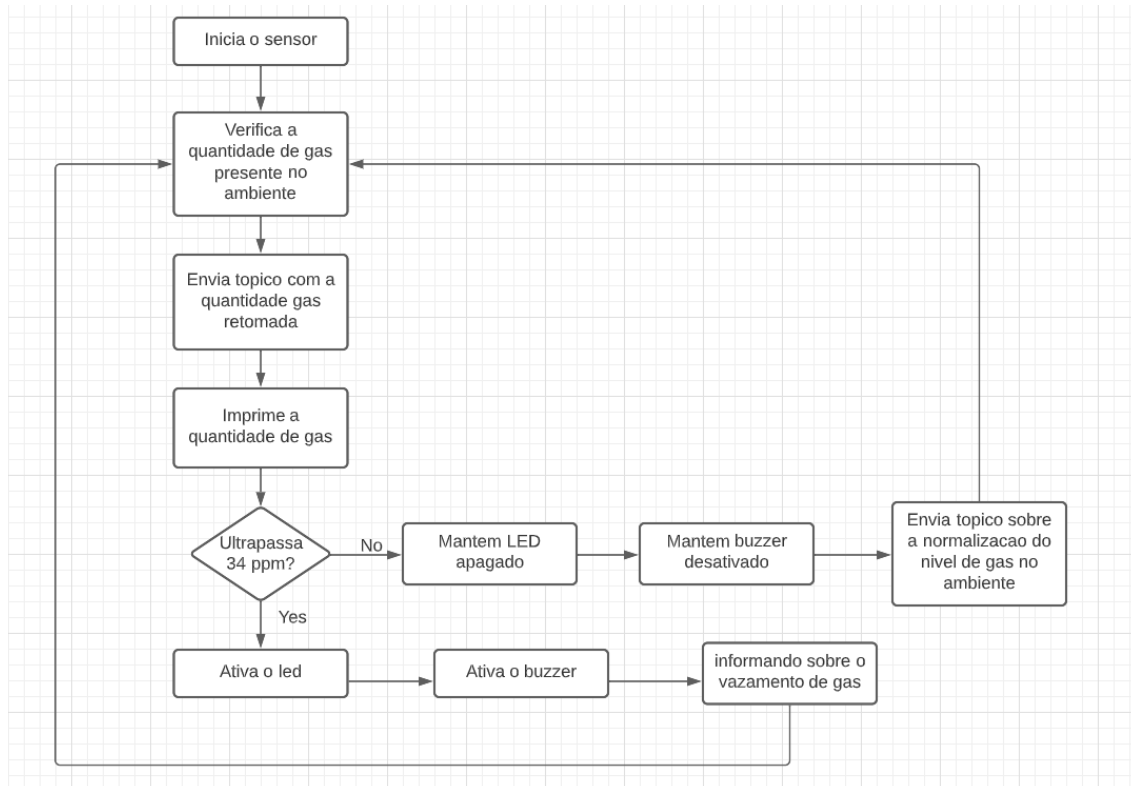


Figura 8. Diagrama de Fluxograma

Fonte: Antonio Mira Gomes Neto

- **3. Resultados**

- **3.1 Circuito de testes**

Primeiramente irei demonstrar como ficou a montagem do nosso projeto:

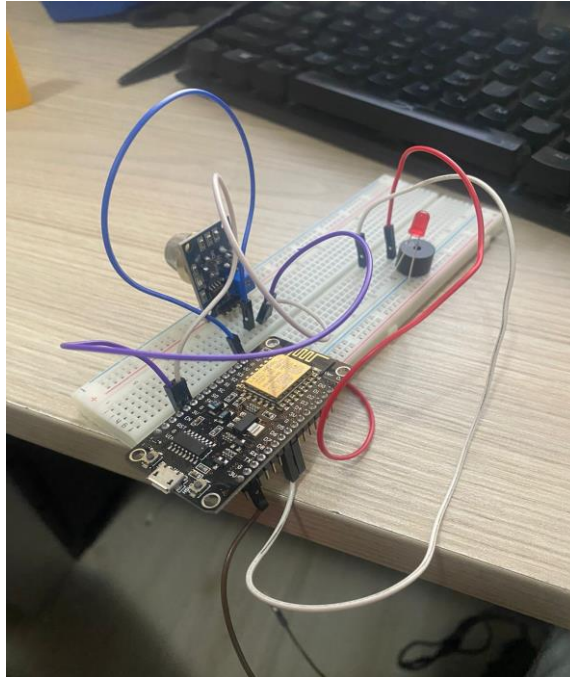


Figura 9. Circuito

Fonte: Antonio Mira Gomes Neto

Após o desenvolvimento do circuito de fomas para o desenvolvimento do código que após vários erros e testes conseguimos concluir e fazer funcionar o sensor e os atuadores, após conseguimos também concluir a parte de mqtt e fomas para parte de Node-RED

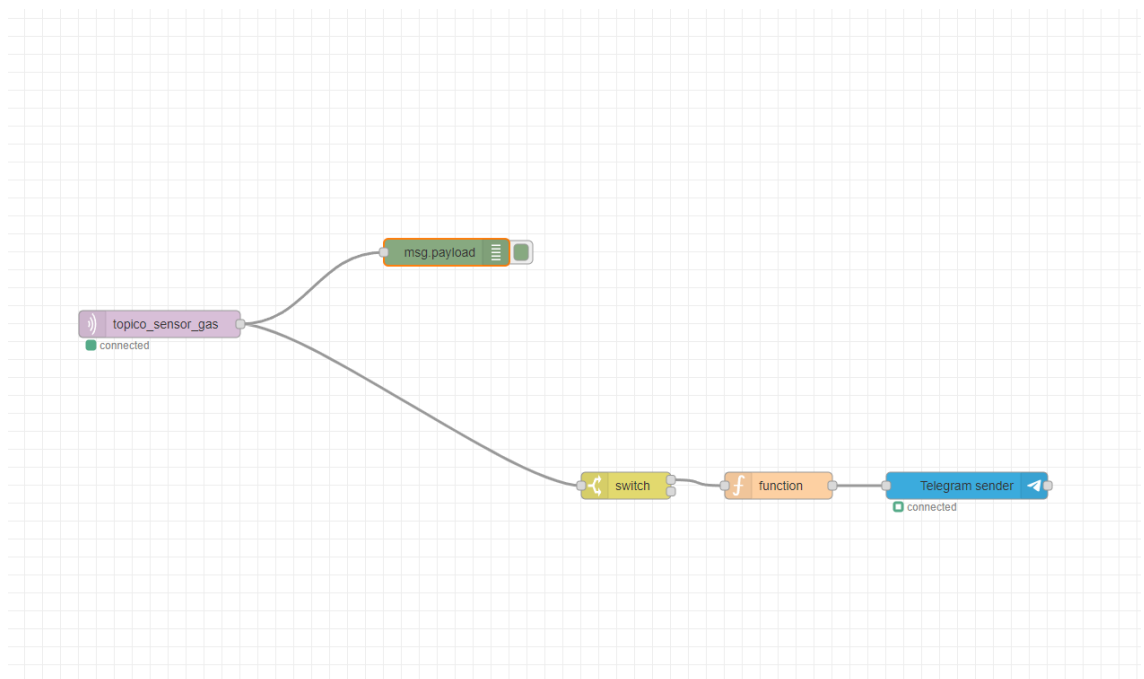


Figura 10. Node-RED

Fonte: Antonio Mira Gomes Neto

Após isso conseguimos conectar o Node-RED com o bot do telegram fazendo ele enviar uma mensagem toda vez que o nível de gás no ar estiver alto

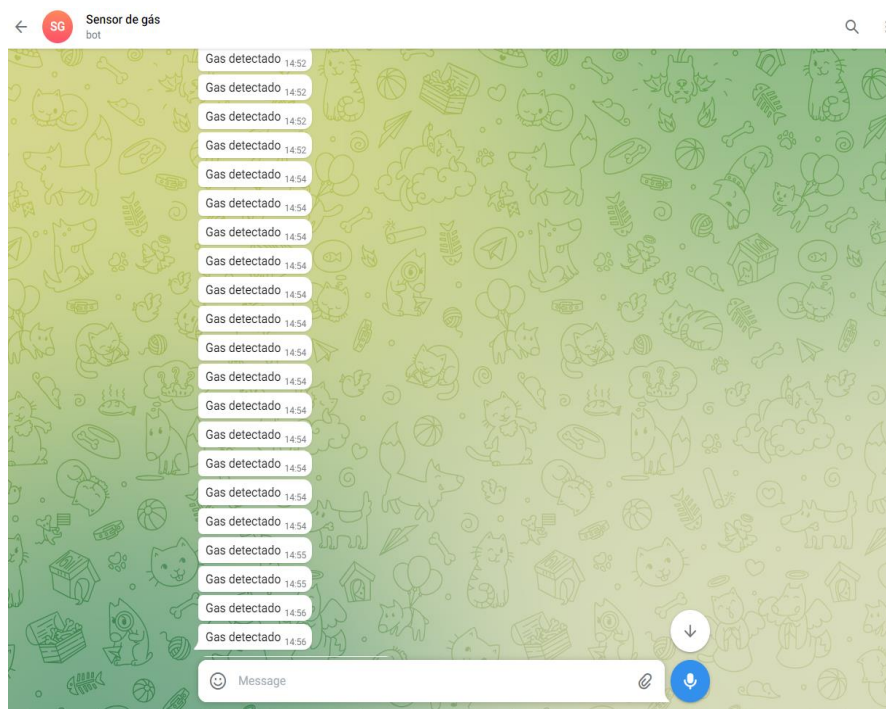


Figura 11. Telegram

Fonte: Antonio Mira Gomes Neto

Ao final, após desenvolvimento e teste, chegamos ao vídeo mostrando resultado do projeto final, disponível em:

Assim como toda a documentação no GitHub

4.0 Conclusões

Os objetivos propostos foram alcançados. No início parecia algo quase impossível, pois após poucos dias de aula com esta nova matéria, recebemos a notícia sobre o projeto e de início fiquei assustado pois não conhecíamos quase nada sobre Arduino. Mas com as aulas, estudos, vídeos e artigos adquiridos consegui atingir o que era esperado e ficamos muito felizes, pois foi uma grande conquista. Além do projeto influenciar em nossos currículos e vidas profissionais, pela sua grandiosidade e importância do projeto. Assim como o objetivo de monitoramento da quantidade de gás no ambiente, e o principal objetivo, influenciado pela necessidade de segurança proposta pelo projeto também foi alcançada, assimilando o aviso para o usuário por meios sonoros e visuais, o que possibilitaria a prevenção de acidentes domésticos e/ou empresariais durante o cotidiano. As vantagens foram várias, porém vamos citar algumas delas. O conhecimento e experiência adquiridos no projeto influenciou muito, pois após a execução do mesmo, notamos a importância não somente do Hardware, mas também o quanto um projeto desses poderia salvar vidas e evitar acidentes. Podemos dizer também que a complexidade foi uma grande vantagem, pois através dela, nos desafiamos e mergulhamos na matéria e no conteúdo para atingir o máximo esperado pelo professor.

Como o Hardware se apresenta totalmente pronto e funcional, poderia ser feito algo relacionado ao aspecto visual, que seria no caso implementar uma carcaça, uma aparência visual, cobrindo assim, os fios e todo hardware. Deixando basicamente pronto para utilizar em um domicílio normal, com uma proteção, uma boa aparência.

5. Referencias

- [1] Oliveira, Greici (2017) “NodeMCU – Uma plataforma com características singulares para o seu projeto IoT”
www.blogmasterwalkershop.com.br/embarcados/nodemcu/nodemcu-uma-plataforma-com-caracteristicas-singulares-para-o-seu-projeto-iot
- [2] Damín, Délio (2019) “Aplicação exemplo: Driver MQTT em comunicação com Broker Mosquitto MQTT.”
www.kb.elipse.com.br/aplicacao-exemplo-driver-mqtt-em-comunicacao-com-broker-mosquitto-mqtt/
- [3] Basílio, Shirley (2021) “O que é Node-RED? Conhecendo e instalando”
www.blogmasterwalkershop.com.br/outros/o-que-e-node-red-conhecendo-e-instalando

Vídeo de implementação do projeto

Sensor de gas, disponível em: <https://youtu.be/3v7nEHRXsnw>

Código utilizado no projeto

Disponível no github: github.com/AntonioNeto122/Sensor-de-gas