



Sensor de gás de cozinha: Evitar acidentes no cotidiano com o uso da IoT

Leonardo Mosca, Victor Enrique, Wilian Franca

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

leoomoscabr@hotmail.com, victorenrique456@gmail.com

Abstract. *This article describes the solution project for gas leaks proposed by the duo, using an MQ-2 sensor and an Internet MQTT protocol that will inform the user on their cell phone about the leak in the form of an alert*

Resumo. *Este artigo descreve o projeto de solução para vazamentos de gás de cozinha proposto pela dupla, utilizando um sensor MQ-2 e o protocolo de internet MQTT que irá informar o usuário em seu celular sobre o vazamento na forma de alerta*

1. Introdução

Apenas em São Paulo, o Corpo de Bombeiros registrou 4.055 casos de acidentes por decorrência de vazamento de gás no ano de 2017, de acordo com artigo (Felipe, Wagner, 2017) [1]. Nosso projeto tem como objetivo atender a necessidade de um risco muito frequente e fatal, que é o vazamento de gás. Através de um pequeno projeto, envolvendo custo benefício, iremos desenvolver um detector de gás, na qual será de fácil acesso e de grande importância, podendo resolver um problema do cotidiano, pelo alto preço de produtos eletrônicos e tecnológicos.

Este projeto tem como objetivo solucionar o problema do vazamento de gás de cozinha, para tentar, diminuir o número de acidentes causados. Para executá-lo iremos desenvolver um software que seja capaz de detectar no ar, gases que excedessem o valor considerado como normal. Assim quando ultrapassados seria transmitido um aviso áudio visual, através de um alarme e um LED, alertando sobre o possível vazamento de gás ou de qualquer químico presente no local, assim como também será enviado um alerta para o usuário pelo seu smartphone.

2. Materiais e Métodos

2.1 Materiais e métodos utilizados

- **NodeMcu ESP8266**

Os módulos utilizados para o desenvolvimento do projeto foram, a development board NodeMcu ESP8266 (Figura 1), onde este é um acoplado de outras duas placas, sendo elas, a NodeMCU Devkit 1.0 e o módulo ESP-12E. O NodeMcu ESP8266 foi utilizado como microcontrolador e principal agente para estabelecimento da comunicação via WiFi do circuito com a rede no sistema, através dele também foi possível o monitoramento da leitura do sensor utilizando o protocolo MQTT, a escolha por este microcontrolador foi devido a sua fama no meio IoT, seu suporte a utilização dos protocolos MQTT e TCP/IP e sua praticidade, assim como citado na publicação sobre a utilização do módulo ESP8266 (Murta, Gustavo, 2018) [3].

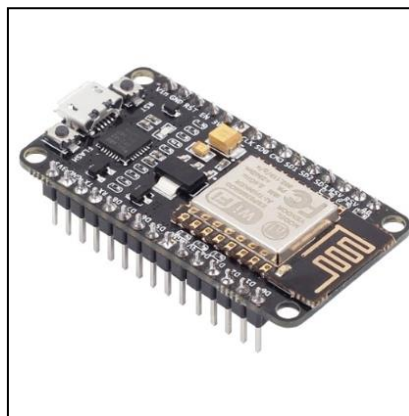


Figura 1. NodeMcu ESP8266

fonte: <https://www.filipeflop.com/produto/modulo-wifi-esp8266-nodemcu-esp-12/>

- **Sensor de gás e fumaça MQ-2**

O sensor MQ-2 (Figura 2) tem a capacidade de detectar a concentração de vários gases combustíveis e fumaça no ambiente, como por exemplo, o Butano, principal componente do gás de cozinha, o sensor também é capaz de captar outros elementos, tais como GLP, Metano, Propano, Hidrogênio, Álcool e Gás Natural, ele capta a quantidade desses elementos através da concentração, em partículas por milhão (ppm) destes presentes no ambiente, sua capacidade de leitura e monitoramento vai de 200 a 10000ppm.

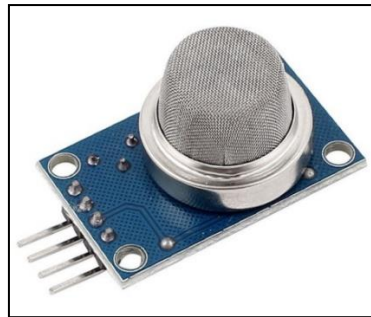


Figura 2. Sensor MQ-2

fonte: <https://www.institutodigital.com.br/pd-141115-modulo-sensor-de-gas-inflamavel-e-fumaca-mq-2.html>

- **Buzzer piezo sonalarme s-30v lf**

Escolhido para ser o atuador do projeto, o Buzzer (Figura 3) tem como principal objetivo no sistema a notificação de um vazamento de gás por meio sonoro para o indivíduo mais próximo, emitindo um som semelhante a um alarme, seu principal funcionamento é devido ao cristal contido em seu nome, o Piezo, o qual pode ser manipulado aplicando alterações na frequência da voltagem que passa pelos fios da peça, gerando assim diferentes efeitos sonoros, conforme citado na publicação sobre seu funcionamento (DiCola, Tony, 2018).



Figura 3. Buzzer Piezo

fonte: <https://proesi.com.br/componentes-eletricos/buzzer-microfone-e-sonalarme.html>

- **Protoboard 830 pinos**

Utilizada como matriz de contato, a protoboard (Figura 4) é uma peça com base em plástico, responsável pela ligação e transmissão dos sinais elétricos entre os componentes do circuito, através de sua função, nela o sensor foi acoplado diretamente, a placa escolhida foi a de 830 pinos, devido a garantia de espaço suficiente para comportar todos os elementos do projeto.

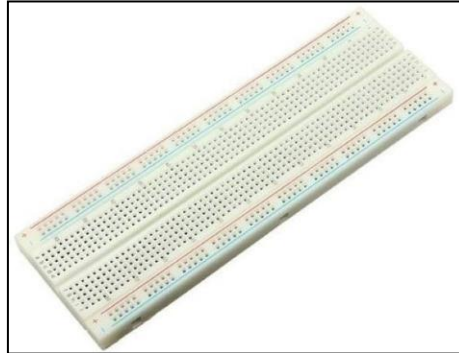


Figura 4. Protoboard 830 pinos

fonte: <https://www.filipeflop.com/produto/protoboard-830-pontos/>

- **LED difuso 5mm vermelho**

Utilizado como indicador visual do projeto, se trata de um pino de LED comum (Figura 5), o qual é ligado a todo o circuito através da protoboard e irá responder aos sinais captados pelo sensor através do processamento lógico do projeto.

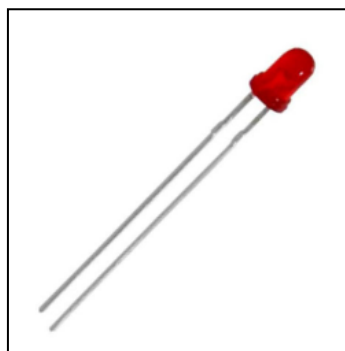


Figura 5. LED difuso vermelho

fonte: <https://natalmakers.lojaintegrada.com.br/led-difuso-5mm-vermelho>

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

Responsáveis por interligar, seja através ou não, da protoboard, os cabos jumper são fios feitos de cobre com pinos de conexão em suas pontas, eles estão disponíveis nas composições: macho/macho, macho/fêmea e fêmea/fêmea, no projeto foram utilizados apenas cabos macho/macho (Figura 7) e macho/fêmea (Figura 6).



Figura 6. Cabos jumper macho/fêmea

fonte: <https://www.curtocircuito.com.br/kit-jumper-macho-femea-40-pcs.html>



Figura 7. Cabos jumper macho/macho

fonte: <https://www.smartprojectsbrasil.com.br/kit-cabo-jumper-macho-macho-10-unidades-20cm>

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.

- **TCP/IP**

O protocolo TCP/IP também se baseia em uma comunicação entre servidor e cliente. Basicamente sem este protocolo dois ou mais computadores não conseguem se comunicar entre si, pois necessitam de uma linguagem padrão, então para isso serve o protocolo TCP/IP. Este protocolo TCP (Protocolo de Controle de Transmissão) tem como função dividir as mensagens que serão enviadas à outra máquina em pequenas mensagens e então encaminha-las via internet. O protocolo que vem acompanhado, o IP (Protocolo de Interconexão) é um guia, no qual fornece o caminho (endereço) da entrega dos pacotes de mensagens.

- **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.

- **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.

- **MQTT Dash**

Aplicativo gratuito para smartphone, compatível com Andoid e IOS, utilizado para criação do Dashboard e visualização/manipulação dos tópicos enviados com a utilização do protocolo MQTT.

Toda a codificação do projeto foi realizada através da Arduino IDE, onde nela foi realizada a integração de todos os serviços e protocolos, começando pelo MQTT, onde foram utilizados tópicos de publish para envio das informações ao usuário, todos estes processados pelo Broker Service, Mosquitto, e por último enviados ao Dashboard criado no aplicativo MQTT Dash, como forma de experiência final para o usuário.

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 ultrapassam 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 outros serviços e dispositivos por meio do protocolo MQTT, conforme mensurado nos artigos sobre definição e manipulação do MQTT, (Oliveira, Bruno, 2017) [2] e (Neri R.; Lomba M.; Bulhões G., 2019) [4].

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.

Após a importação das bibliotecas são declaradas as variáveis com as informações para autenticação na rede WiFi assim como a constante que do tipo WiFi, que será utilizada pelos métodos da biblioteca PubSubClient para manipulação das informações através dos métodos de publish e subscribe. Após a declaração das variáveis é feita a definição dos principais métodos de conexão aos principais componentes que estabelecerão a comunicação entre dispositivos diferentes, sendo o WiFi e o MQTT Broker, respectivamente

Os níveis de gás no ambiente serão constantemente capturados pelo sensor MQ-2, e um retorno contendo a quantidade em partículas por milhão (ppm) é exibida no monitor serial, assim como armazenada em uma variável que será utilizada mais tarde no código.

São então definidos os pontos de conexão, entradas analógicas e digitais que foram estabelecidas no circuito físico, assim como a definição de suas funções, juntamente são também iniciadas a taxa de transferência em bits por segundo para transmissão serial, o estabelecimento do server e porta que serão utilizados pelo MQTT, assim como a chamada da função de call-back que irá ser responsável por construir a relação para recebimento do tópico que será enviado e impressão das informações contidas no payload.

Por último, na função de loop, é onde todo o processamento das informações ocorre, primeiramente é feita a declaração das variáveis char que poderão compor conteúdos de payloads, que por sua vez poderão ser enviados junto aos tópicos através dos métodos publish ou subscribe, logo após ocorre a chamada da função que irá tentar estabelecer a conexão com o WiFi e o MQTT Broker Service, após o estabelecimento das conexões é feita a leitura dos dados obtidos pelo sensor através da porta analógica A0, onde estas são exibidas no monitor serial, e também através do método sprintf(), armazenadas concatenadas ao array de char responsável pelo armazenamento da quantidade, em partículas por milhão, de gás presente no ambiente, a qual será enviada por meio de um dos tópicos

Após a captura desse valor, é estabelecida uma estrutura condicional, a qual irá verificar a quantidade de gás presente no ambiente e comparar com um determinado valor, caso esta ultrapasse o valor estabelecido, ocorre o acionamento de uma série de ações, sendo o acionamento do LED, a chamada do método publish para o tópico que irá notificar o usuário através do Dashboard sobre o vazamento de gás utilizando um array de char com uma mensagem como payload e o acionamento do buzzer, respectivamente.

Caso contrário, o LED irá se apagar, a mensagem de alerta será alterada, também através do método publish, por uma mensagem informando um estado de normalização do gás presente no ambiente e o buzzer também será desligado, respectivamente, todo esse processo para envio das informações será em função de um delay de 1 segundo, estabelecido ao final da estrutura da função de loop.

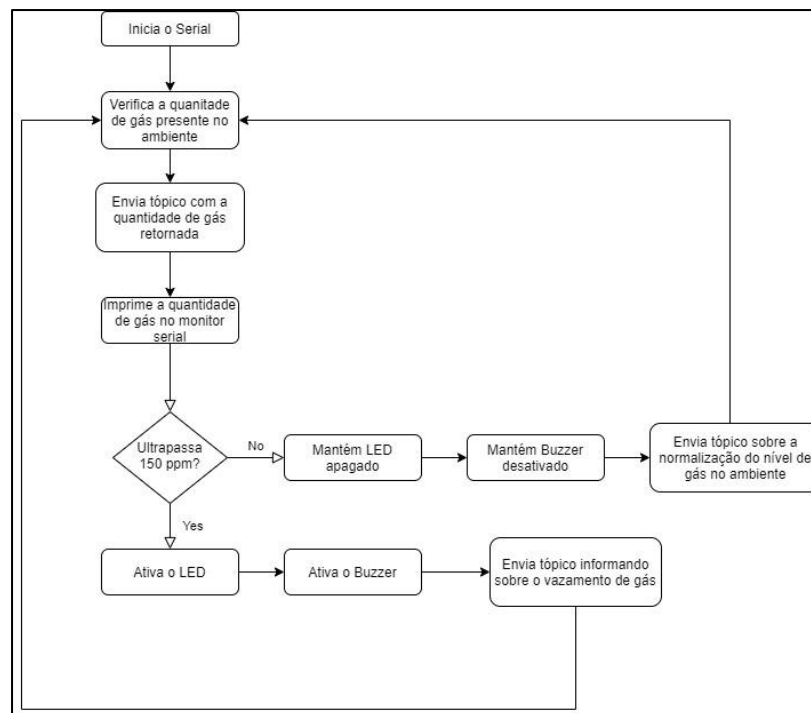


Figura 8. Fluxograma
fonte: (Victor Enrique Marinho Caetano, 2020)

3. Resultados

O produto final tem como objetivo garantir a segurança assim como o bem-estar das pessoas, com o intuito de prevenir possíveis acidentes no cotidiano, alertando-as de vazamentos de gases os quais são bastante perigosos e causadores de uma série de acidentes. Lembrando que este produto detecta tanto gases quanto fumaça, podendo assim ser utilizado próximo de fornos elétricos, torradeiras, sanduicheiras, além de fogões.

Ele é capaz de detectar diversos tipos de gases e fumaças. O sensor recebe informações sobre o nível de gases e fumaça presente no ar, obtendo esses dados em partículas por milhão (ppm), no qual o valor será armazenado em uma variável. E este será comparado a um valor pré-estabelecido em um estrutura condicional, onde caso atinja esse valor, acionará o restante das funcionalidades do sistema.

Sendo assim ele irá comparar o nível de gás presente no ambiente com o limite estabelecido no código, caso ultrapassado ele ativará um aviso audiovisual através de um LED, um alarme e a notificação pelo Dashboard de monitoramento, alertando da detecção de vazamento de gás e prevenindo possíveis acidentes.

Mais detalhes sobre o código desenvolvido, do circuito eletrônico e de todo o projeto podem ser encontrados na apresentação localizada no youtube, acessível através do endereço <https://www.youtube.com/watch?v=t-z4G-xYI-g> [5], assim como toda a documentação referente ao projeto pode ser acessada através do repositório GitHub <https://github.com/Victor-182/SensorGasInteligente> [6].

Um dos problemas iniciais foi pensar em uma forma para realização dos testes de leitura dos dados pelo sensor, tendo em vista que o botijão de gás é uma fonte de matéria-prima do perigosa quando manipulada fora de um ambiente controlado, com isso foi realizada uma pesquisa sobre o funcionamento do sensor MQ-2, onde foi verificado, que uma fonte menor do gás poderia ser utilizada para realização dos testes, sendo esta proveniente de um isqueiro, em que seu fluido é composto por: Butano (C_4H_{10}) Nafta (C_6H_6) Querosene.

Uma das maiores dificuldades presentes durante o desenvolvimento do projeto, foi o fator singular relacionado a pandemia de COVID-19, devido ao distanciamento social obrigatório, toda a parte de hardware teve que ser desenvolvida apenas por um componente do grupo, havendo auxílio apenas remoto da outra parte, porém, isso foi contornado com o planejamento e divisão das tarefas, onde um dos componentes do grupo ficou responsável por toda a documentação e desenvolvimento do artigo, enquanto a outra ficou responsável pelo desenvolvimento do hardware e código do projeto, onde ao final houve a necessidade de integração entre ambas as partes, para preenchimento de lacunas no artigo e um maior acordo entre o funcionamento do projeto físico e a proposta que havia sido estabelecida.

4. 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 ficamos assustados pois não conhecíamos quase nada sobre Arduino. Mas com as aulas, estudos, vídeos e artigos adquiridos conseguimos 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.

Entre as desvantagens temos um imprevisto ocorrido que foi a epidemia causada pelo COVID-19, no qual influenciou bastante pois afastou tanto os alunos quanto os professores, colocando as partes envolvidas afastadas e dificultando algumas atividades. Pelo fato do projeto ser desenvolvido fisicamente, tivemos que dividir algumas tarefas até mesmo para manter a distância social estabelecida pela OMS (Organização Mundial da Saúde) no qual afetou um pouco o desenvolvimento.

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.

Juntamente a carcaça, poderíamos também incluir um método de proteção/prevenção adicional, implementando mais um atuador, além do Buzzer, como um braço mecânico, por exemplo, que poderia ser acoplado perto do componente que abre e fecha a passagem do gás de cozinha entre o botijão e o fogão para que os piores cenários sejam evitados de forma automática também.

5. Referências

- [1] Wagner, Felipe (2017) “Estatísticas Impressionantes – Histórico De Acidentes Com Gás”,
<https://www.rwengenharia.eng.br/acidentes-com-gas>, Julho.
- [2] Oliveira, Bruno (2017) “Dando uma breve análise no protocolo MQTT“,
<https://medium.com/internet-das-coisas/iot-05-dando-uma-breve-an%C3%A1lise-no-protocolomqtt-e404e977fbb6>, Setembro
- [3] Murta, Gustavo (2018) “Guia completo do NodeMCU – ESP12 – Usando Arduino IDE”, <https://blog.eletrogate.com/nodemcu-esp12-usando-arduino-ide-2/>, Março
- [4] Neri, R.; Lomba, M.; Bulhões, G. (2019) “MQTT”,
<https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/vf/mqtt/>, Janeiro
- [5] Enrique, Victor; Mosca, Leonardo (2020) “Sensor de Gás Inteligente - Objetos Inteligentes Conectados”, <https://www.youtube.com/watch?v=t-z4G-xYI-g>, Junho
- [6] Enrique, Victor; Mosca, Leonardo (2020) “SensorGasInteligente”,
<https://github.com/Victor-182/SensorGasInteligente>, Junho