



Projeto Integrado em Computação I

Sistema Inteligente Doméstico para Detecção e Alerta de Vazamento de Gás

Bernardo Amorim
Mirelly Micaela
Victoria Louzada

Trabalho da disciplina de Projeto Integrado
de Computação I apresentado na UFES
no dia 05 de dezembro de 2019.
Professor: Vinicius Fernandes Soares Mota

1- Objetivo

A ideia de implementar um projeto no qual a funcionalidade principal é identificar uma possível exposição de gás e alertar um usuário em questão se dá pelas inúmeras ocorrências anuais de acidentes domésticos causados por vazamentos que, na maioria das vezes, passam despercebidos pelas pessoas.

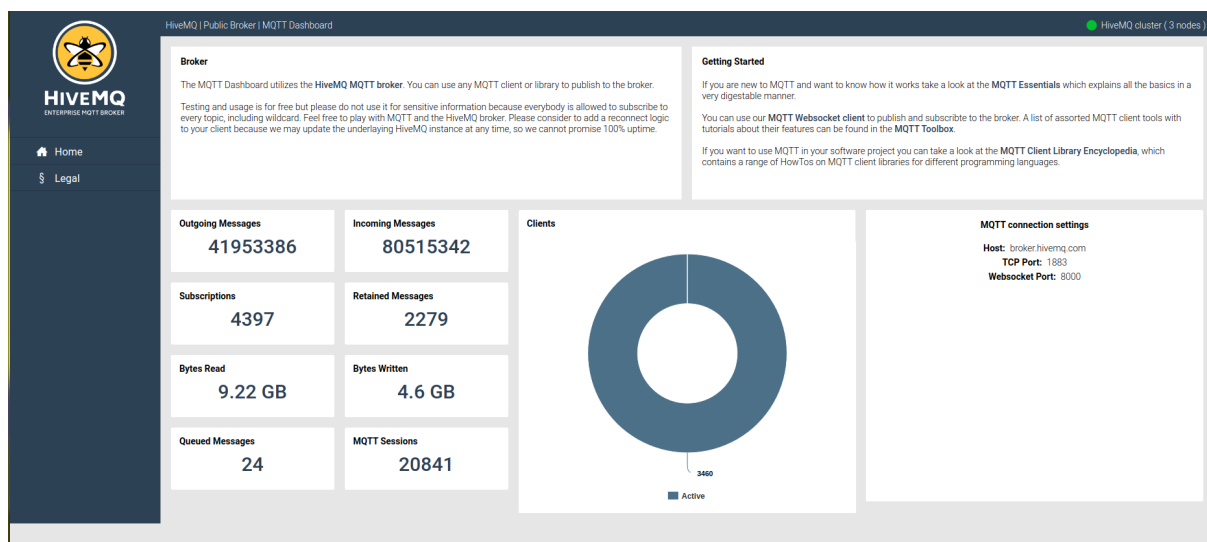
2 - Materiais utilizados

Hardware:

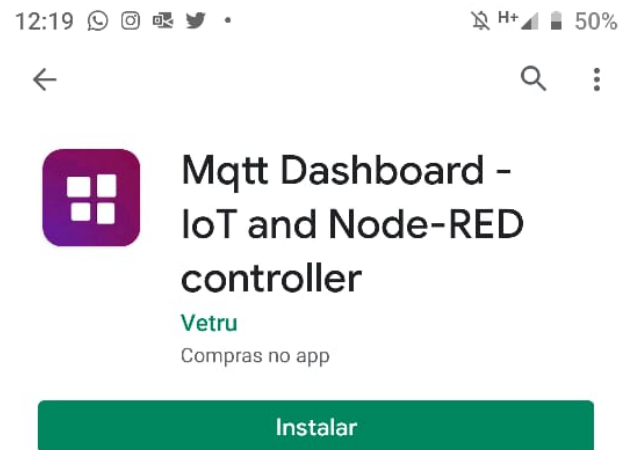
- Arduino Uno
- ESP8266
- Buzzer
- Sensor de gás MQ-9
- Display LCD 16x2
- Jumpers
- Protoboard

Software:

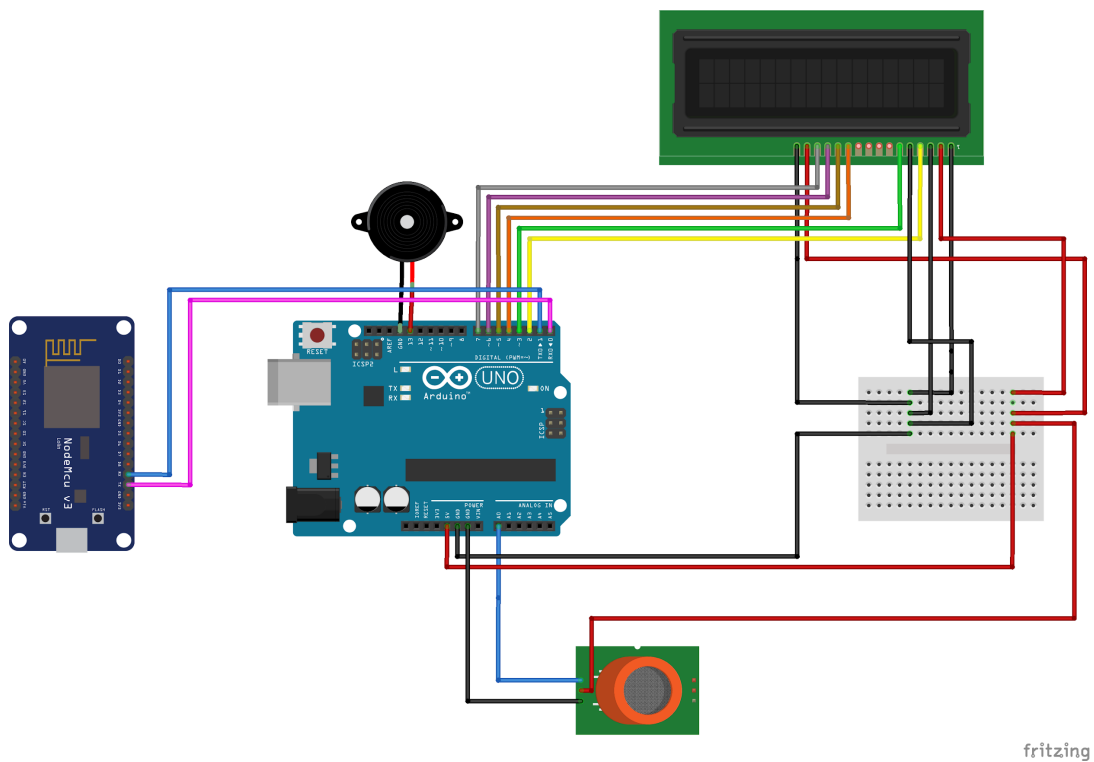
- Site Servidor Mqtt: <http://broker.mqtt-dashboard.com/>



- Aplicativo Mqtt Dashboard, disponível na PlayStore.



3 - Montagem do circuito



Vitória - ES
2019

4 - Funcionamento

Ao ligar o circuito é dado um delay de 1 minuto, que é o tempo suficiente para que o sensor MQ-9 aqueça e estabilize parcialmente seus valores de leitura, e para que o ESP8266 faça as conexões com o Wifi e o servidor mqtt. No servidor mqtt, é criado um tópico chamado PIC20192. O aplicativo Mqtt Dashboard está assinado neste tópico, assim qualquer mensagem publicada nele poderá ser vista no app.

Após isso, o display exibe a mensagem “Calibrando” durante mais 1 minuto. Neste tempo, o programa está recebendo os valores de leitura do MQ-9 e fazendo uma média para saber qual é o valor padrão da quantidade de gás no ambiente. É definido como limite o valor de 50% acima da média.

Após a calibração, o sistema está pronto para ser usado.

Enquanto circuito estiver ligado, a cada 1 segundo:

- no display é exibida a mensagem “GÁS AUSENTE” , o valor limite e o valor de leitura atual;
- o sensor MQ-9 faz uma leitura do ar e verifica se o valor está acima da média. Caso o valor lido esteja acima da média:
 - 1) O buzzer é ativado;
 - 2) No display é exibida a mensagem “GÁS DETECTADO!”;
 - 3) O Arduino, através da porta TX, envia uma mensagem para o ESP8266 que, ao recebê-la, publica no tópico ‘PIC20192’ a mensagem “GÁS DETECTADO!”. O aplicativo Mqtt Dashboard exibe na tela esta mensagem, desta maneira o usuário consegue ser avisado imediatamente se há vazamento de gás.

Caso, o valor volte a estar abaixo da média, a mensagem para de ser publicada, o display exibe “GÁS AUSENTE” e o buzzer é desativado.

5 - Experiência

O projeto inicial consistia em um sistema independente feito a partir de um módulo de ESP32, capaz de detectar a presença de gás inflamável e, ao fazê-lo, acionar um buzzer local, enviar uma mensagem para o telefone do usuário cadastrado através do servidor MQTT, utilizar um servo motor para fechar a válvula do gás e, sendo assim, interromper o escapamento. Entretanto, fizeram-se necessárias alterações constantes devido aos entraves encontrados.

Logo nas primeiras semanas de teste, nosso foco principal era implementar de maneira correta o sensor MQ-2 de gases combustíveis e fumaça alinhado ao buzzer. Contudo, observamos algumas dificuldades à medida que realizávamos os testes e foi preciso trocar o sensor, já que o primeiro começou a apresentar erros na análise de valores médios do gás em ambiente normal e com presença do tipo tóxico.

Após solucionar esse primeiro problema, o dispositivo de gás e o alarme funcionavam de maneira harmônica e, então, partimos para o próximo passo: acoplar o servo motor ao nosso pequeno circuito.

Para adicionar um motor do tipo servo a um conjunto desse modelo, é necessário incluir no código uma biblioteca contendo funções específicas do servo. Esse foi o primeiro problema acerca dele com o qual nos deparamos. A placa ESP32 que utilizávamos, até então, não permitia o funcionamento dessas funções. Esse seria um dos motivos que viria nos obrigou a adaptar nossos planos e trocar o módulo ESP onde o programa rodava.

Paralelo a isso, o servidor com o qual nosso sistema se comunicaria estava sendo preparado também. Determinamos as mensagens que seriam enviadas ao usuário, parâmetros para isso acontecer, qual rede de wi-fi estaria conectada à placa, mas quando o código foi compilado para ser transferido para o microcontrolador, descobrimos que o tipo de conexão que queríamos não era compatível com ele. Esse acontecimento foi o decisivo para trocar para o ESP8266.

Ao mudar o dispositivo base do nosso programa, tivemos mais algumas surpresas:

- 1) o ESP não cabia na protoboard e não conseguimos uma maior que atendesse as necessidades;
- 2) por causa disso, seria preciso acoplar e utilizar apenas um lado da placa na protoboard, mas isso acarretou na próxima dificuldade;
- 3) não tinham saídas digitais e analógicas suficientes para atender nosso circuito.

4) mesmos as operações mais básicas não rodavam.

Acabamos por descobrir que o problema estava no microcontrolador e trocamos apenas de módulo, mantendo o modelo 8266, mas isso só resolveu o último problema.

Ao explicar a situação para o professor, ele orientou que trocássemos novamente de placa, adotando um Arduino UNO como dispositivo principal que controlaria as ações do sensor, buzzer e servo motor e a placa do ESP8266 apenas como módulo wi-fi para a comunicação remota, considerando que o Arduino não possui tal funcionalidade - observe que este modelo resolveu os eventuais problemas em relação à conexão com o MQTT .

Após montar o sistema utilizando os novos dispositivos, um problema persistiu: o mau funcionamento do servo motor que acarretava em erros em todos os outros aparelhos. Por causa disso, chegamos ao consenso de que retirar a funcionalidade do fechamento da válvula seria o mais prudente, já que, mesmo com todos os testes, principalmente aqueles nos quais apenas o servo era empregado, não foi possível corrigir os erros. Segundo alguns artigos na internet, uma provável causa para tais erros seria o fato do Arduino não fornecer a energia necessária para o pleno funcionamento dele fazendo com que a placa reiniciasse ou, até mesmo, roubasse energia das outras portas seriais.

Visto que eliminamos uma função do nosso projeto, precisávamos substituí-la e decidimos incrementar um visor LCD 16x2. Nele seriam impressas todas as mensagens de calibragem e detecção de gás. Não tivemos problemas em instalá-lo haja vista que seria necessário apenas transferir do monitor de saída serial para a saída do LCD as mensagens do código.

Apesar do trabalho estar completo, códigos compilados, todos os testes possíveis realizados, todas as funções produzem com excelência suas tarefas individualmente, a comunicação do Arduino com o ESP e MQTT ainda é uma função instável no programa e, quando há problemas com o ESP, esses são refletidos no andamento de todo o projeto já que há uma conversação direta entre esses dois microcontroladores.

Felizmente, nosso projeto é montado de maneira que, caso seja julgado melhor eliminar a comunicação remota, isso não afeta o andamento do resto do sistema. Portanto, os entraves que lidamos hoje são de direta solução. Sendo assim, tudo que é necessário para que a execução ocorra de maneira eficiente é uma fonte de energia estável e, no máximo, uma rede de wi-fi acessível.

Independente de todas as dificuldades enfrentadas ao longo das semanas de execução, a elaboração desse projeto foi uma experiência de grande aprendizado para todo o grupo, dado que um processo de criação ou de experimento científico é baseado na ideia de erros e acertos na qual somente assim é possível mapear todas as nuances de tal situação a ser testada. Adicionado ao fato de ser o primeiro contato de praticamente todos os envolvidos com as plataformas de desenvolvimento em questão, as pesquisas necessárias para resolver os contratempos, se tornaram consideravelmente proveitosas.