

Sistema de Monitoramento de Qualidade de Ar

Adolfo de Souza Serique, 140127640
Universidade de Brasília, Faculdade Gama
Sistemas Embarcados, 206181
Brasília, Brasil
adolfoferique@gmail.com

Universidade de Brasília, Faculdade Gama
Sistemas Embarcados, 206181
Brasília, Brasil
luanacarolinadva@gmail.com

Luana Carolina de Val Abreu, 140150234

Resumo — Sistema baseado em localização, medição dos gases atmosféricos e temperatura via sensores que fornecerão uma estimativa da qualidade do ar naquele determinado local.

I. INTRODUÇÃO

A poluição do ar é um dos maiores desafios ambientais e de saúde pública no mundo atualmente. A poluição do ar leva a efeitos adversos na saúde humana, clima e ecossistema. Estudos indicam que a exposição de crianças a gases nocivos podem causar danos ao seu desenvolvimento [1]. O ar vem sendo poluído por causa da liberação de gases tóxicos pelas indústrias, emissões veiculares e aumento da concentração de gases nocivos e material poluente na atmosfera.

Frequentemente, os efeitos da má qualidade do ar não são tão visíveis comparados a outros fatores mais fáceis de serem identificados. Contudo, os estudos epidemiológicos [2] têm demonstrado, correlações entre a exposição aos poluentes atmosféricos e os efeitos de morbidade e mortalidade, causadas por problemas respiratórios (asma, bronquite, enfisema pulmonar e câncer de pulmão) e cardiovasculares. As populações mais vulneráveis são as crianças, os idosos e as pessoas que já apresentam doenças respiratórias.

Com a evolução da tecnologia como a Internet das Coisas e Raspberry Pi, é mais fácil implantar sensores que permitem a detecção da qualidade do ar em tempo real. A Internet das Coisas ou *Internet of Things (IoT)*, em inglês, é um termo utilizado para descrever um paradigma tecnológico no qual os objetos físicos estão conectados em rede e são acessados através da Internet. A integração de IOT, sensores e com a ajuda de Raspberry Pi para monitoramento da qualidade do ar fornece uma maneira eficaz do que as abordagens que as usadas anteriormente.

A meta desse sistema é ler dados sobre poluição na atmosfera em tempo real. Para isso é necessária uma estrutura de hardware e software. Sensores de temperatura, localização e de gases serão responsáveis

por ler os dados, e esses requerem uma interface que os lê, interpreta e exibe.

II. DESENVOLVIMENTO

Para identificar os níveis de gases poluentes presentes na atmosfera será utilizado o sensor MQ135. O Sensor de Gás MQ-135 para gases tóxicos é um módulo capaz de detectar vários tipos de gases tóxicos como amônia, dióxido de carbono, benzeno, óxido nítrico, e também fumaça ou álcool.

Existe uma grande gama de sensores de gás, entre eles, temos os MQ's. Esses sensores de detecção modernos e de baixo custo para gás combustível, são do tipo eletro-catalítico. São feitos de uma bobina de fio de platina aquecido eletricamente, coberto primeiro com uma base de cerâmica, tais como alumina e, em seguida, com um revestimento final exterior do catalisador de paládio ou ródio disperso em um substrato de tório.



Figura 1: MQ135

Como esse sensor não possui um conversor AD acoplado, será necessário a implementação de um dispositivo que cumpra essa missão. O MCP3008 é um conversor AD de 8 canais, com 10 bits de resolução com um período de amostragem aceitável para o projeto em questão.

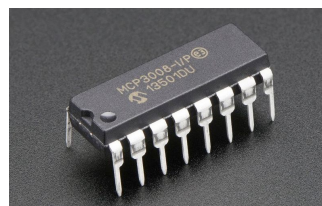


Figura 2: MCP3008

Para controle da temperatura e umidade do ar, será implementado o AM2302. É um sensor com baixo consumo de corrente e que possui internamente um sensor de umidade capacitivo e um termistor, além de um conversor analógico/digital para comunicação com o microcontrolador. Este nos permite fazer leituras de temperaturas entre -40 a +80 graus Celsius e umidade entre 0 a 100%.

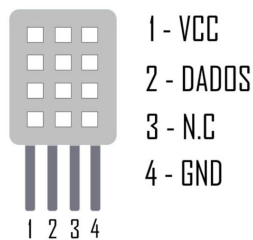


Figura 3: AM2302

Por fim, será utilizada a Raspberry Pi para realizar o processamento dos dados e a conexão dos mesmos com um servidor, que os armazenará e apresentará de forma intuitiva ao usuário

III. HARDWARE

Utilizou-se o Fritzing para realizar o esquemático do hardware. A Raspberry está conectada diretamente no pino 4 GPIO e no pino que será definido para o MQ135. Utilizamos também os pinos que fornecem 3.3V e GND.

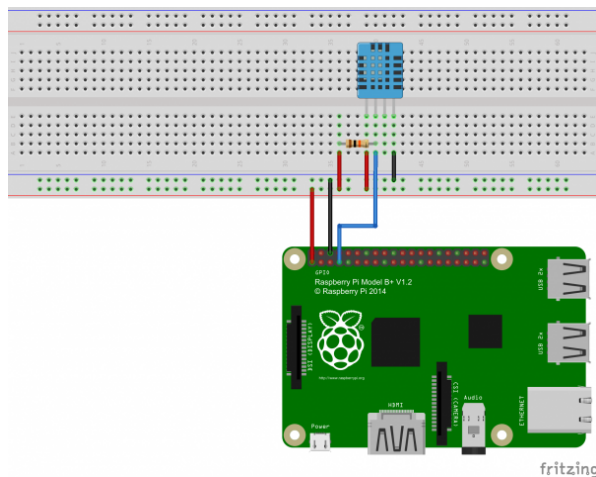


Figura 4: Esquemático para o AM2302

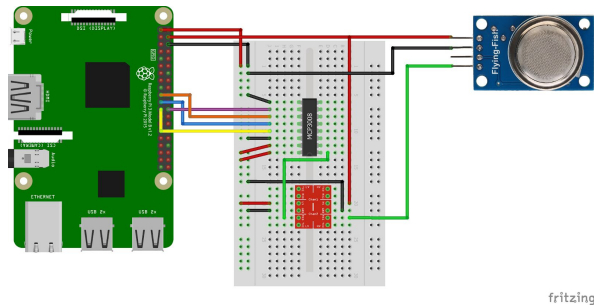


Figura 5: Esquemático para o MQ135 e MCP3008

IV. DESCRIÇÃO DE SOFTWARE

A. Configurações iniciais

Antes de iniciar a programação dos componentes, foi necessária a instalação de algumas bibliotecas e recursos, além da configuração de data e hora na Raspberry para manter o servidor sincronizado.

Primeiro certificamos de que os dados da Raspberry estão atualizados com o comando:

```
$ sudo apt-get update
```

Depois deve-se instalar os seguintes recursos:

- Python para rodar o script HTML

```
$ sudo apt-get install -y build-essential python-dev git
```

- SQLite

```
$ sudo apt-get install sqlite3
```

- Apache

```
$ sudo apt-get install apache2
```

- Configuração de data e hora

```
$ sudo apt-get install ntpdate
```

```
$ sudo service ntpd stop
```

```
$ sudo ntpdate time.nist.gov
```

```
$ sudo service ntpd start
```

```
$ sudo cp /usr/share/zoneinfo/Etc/GMT-3 /etc/timezone
```

Com as instalações feitas, pode-se dar continuidade a programação da página HTML e dos sensores.

B. AM2302

A Raspberry coleta dados do sensor pelo pino 4 da GPIO, Para ler os dados usamos as bibliotecas em C chamadas Adafruit_Sensor.h e DHT.h.

```
$ mkdir -p /home/pi/sources
$ cd /home/pi/sources
$ git clone https://github.com/adafruit/DHT-sensor-library
```

Depois de instalada as bibliotecas fizemos um primeiro teste para verificar o funcionamento do sensor.

```
$ sudo
$ /home/pi/sources/DHT-sensor-library/example/AdafruitDHT.c
```

O primeiro argumento é o tipo do sensor que pode ser 11 ou 22. O seguinte é o pino RPi GPIO que está conectado ao pino de dados do sensor. O output depois de executado esse exemplo foi a umidade e temperatura da hora e local em que se realizou o teste.

Temp = 25.3°C Humidity= 26.3%

C. MQ135

Como o projeto em questão não se trata de um sistema crítico logo a sua resposta em real time precisão não é necessária. No caso desse sensor e o conversor AD MCP3008 optamos por usar um código em Python que mesmo sendo linguagem interpretada atende muito bem os requisitos do sistema.

No código em anexo podemos ver que é feito uma calibração do sensor e depois a coleta de dados.

D. Database

É importante salientar que houve uma mudança nesse aspecto do projeto. Substituímos a MySQL pela SQLite por sua simplicidade de instalação e utilização.

Para configurar a base de dados primeiros instalamos o sistema SQLite3.

```
$ sudo apt-get install sqlite3
```

Depois de instalado o sistema, criamos uma base de dados, em seguida criamos uma table com três parâmetros o primeiro do tipo DATETIME que vai relacionar a hora e o dia do dado mensurado, o segundo e o terceiro do tipo NUMERIC relacionados a temperatura e a umidade respectivamente.

```
$ sqlite3 templog.db
$ sqlite BEGIN;
CREATE TABLE temp_hum (timestamp
DATETIME, temp NUMERIC, hum NUMERIC);
```

COMMIT;

E. Servidor

Afim de criar um servidor usou-se a Apache, que é um servidor HTTP de código-livre. Primeiramente instalamos o servidor.

```
$ sudo apt-get install apache2
```

Depois é possível ativar ou desativar o servidor pelos seguintes comandos:

```
$ sudo /etc/init.d/apache2 start
$ sudo /etc/init.d/apache2 stop
```

Para alterar a interface do servidor é preciso modificar seu index.html que configura o que o servidor exibirá em um navegador web.

Para acessar a página contida no servidor basta digitar o ip da Raspberry (no caso 192.168.0.23) em um navegador web.

F. Interface

Utilizou-se python script para alterar a interface do servidor. Inicialmente ela apresenta um gráfico ao longo do tempo (que pode ser definido pelo usuário) da variação da temperatura e umidade. Com o uso do cursor é possível visualizar as medidas pontuais no dia e hora desejados.

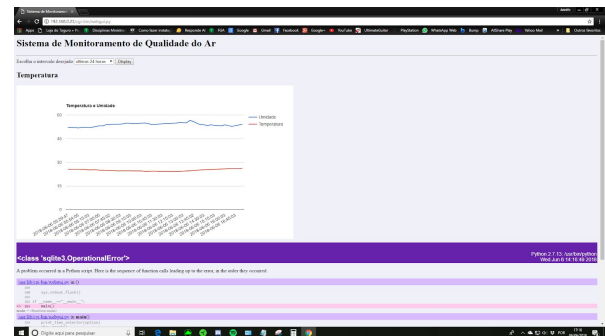


Figura 6: Servidor com os dados do AM2302

IV. CONCLUSÃO

A partir dos argumentos apresentados, a maior dificuldade apresentada pelo projeto é a criação do banco de dados num servidor de forma a apresentá-los de forma intuitiva ao usuário.

Ainda enfrentamos dificuldades para adicionar o sensor de gás na interface do servidor, que será resolvido para o próximo ponto de controle, junto com uma estrutura para comportar o circuito.

V. REFERÊNCIAS

[1] Health Damages from Air Pollution in China, https://dspace.mit.edu/bitstream/handle/1721.1/61774/MITJSPGC_Rpt196.pdf%3Bjsessionid%3D837D627A33DDEFC9B77B8D8C164FA3BBE?sequence%3D1

[2] Ministério de Meio Ambiente. **Qualidade do ar**. Disponível em: <<http://www.mma.gov.br/cidades-sustentaveis/qualidade-do-ar>>

[3] Vida de silício. **Sensor de gás MQ135 e a família de sensores MQ**. Disponível em : <<https://portal.vidadesilicio.com.br/sensor-de-gas-mq-135/>>

[4] <https://tutorials-raspberrypi.com/configure-and-read-out-the-raspberry-pi-gas-sensor-mq-x/>

VI. ANEXOS

Os códigos de encontram na pasta do git:

https://github.com/luanabreu/Sistemas-embarcados/tree/master/projeto_final