

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
ENGENHARIA MECATRÔNICA

EDUARDO HENRIQUE DE SOUZA PIRES

**PROJETO DE UMA UNIDADE DE MONITORAMENTO E CONTROLE
AMBIENTAL**

TRABALHO DE CONCLUSÃO DE CURSO

UBERLÂNDIA - MG
2018

EDUARDO HENRIQUE DE SOUZA PIRES

**PROJETO DE UMA UNIDADE DE MONITORAMENTO E CONTROLE
AMBIENTAL**

Trabalho de Conclusão de Curso apresentado ao
Curso de Engenharia Mecatrônica da Universidade
Federal de Uberlândia como pré-requisito para a
obtenção do Título de Bacharel em Engenharia
Mecatrônica.

Orientador: Professor Doutor José Francisco
Ribeiro

UBERLÂNDIA – MG

2018

RESUMO

A crescente utilização de gases como fonte combustível alternativa, como o gás liquefeito de petróleo (GLP), o gás natural e o hidrogênio, levanta questões quanto a segurança de seu armazenamento e utilização. Este projeto teve como objetivo desenvolver um sistema programável de baixo custo para a detecção de gases e o monitoramento de sua concentração no ambiente. Para tanto, utilizou-se a plataforma Arduino®, sensores da família MQ-x e uma interface LCD. O trabalho discute as condições de operação e também propõe a uma estratégia para a calibração do sensor. Uma vez projetado o sistema, ele foi construído e avaliado e, para tanto, precisou-se realizar alguns testes preliminares, com a finalidade de determinar a resistência *ohmica* dos sensores quando sujeito a um ambiente sem gases. Também foi realizado a captura da curvas de operação dos sensores presentes nos *datasheet* - elementos gráficos – que foram aproximadas por polinômios, tendo em vista a automação da leitura via o Arduino®. O aparato projetado mostrou-se operacional e funcionalmente eficiente no tocante à detecção dos gases no ambiente. A medida da concentração de gás, no entanto, não pode ser adequadamente caracterizada, em função das dificuldades encontradas de se criar um ambiente de teste de referência, ou seja, não se conseguiu ao longo do trabalho – por razões de custo e tempo - submeter o sensor a um ambiente certificado, onde a concentração de gás fosse conhecida com precisão e que possibilitasse confrontar o resultado conhecido do ambiente de teste com as informações oferecidas pelo aparato proposto. Ainda assim uma avaliação funcional foi feita a partir da construção de um ambiente de teste envolvendo um botijão de gás de cozinha, sua válvula reguladora e em vasilhame de 20 litros, onde o sensor foi inserido. Os resultados auferidos, os desafios encontrados e algumas possibilidades de contorno são discutidas ao longo do trabalho. Não obstante as limitações apontadas na caracterização do sensor, o trabalho conclui pela exequibilidade e o potencial do sistema proposto, onde as características de versatilidade, automação e o baixo custo se destacam. O trabalho aponta também para novos estudos e desdobramentos que podem contribuir para a produção de um produto inovador e com um apelo comercial promissor.

Palavras-chave: Arduino. Sensor de Gás. Sistema de Medição.

ABSTRACT

The increasing use of alternative source gas, such as liquefied petroleum gas (LPG), natural gas and hydrogen, raises questions about the safety of its storage and use. This project aimed to develop a programmable low-cost system for the detection of gases and the monitoring of their concentration in the environment. For that, the Arduino® platform, MQ-x family sensors and an LCD interface were used. The paper discusses the operating conditions and also proposes a strategy for the calibration of the sensor. Once the system was designed, it was constructed and evaluated, and it was necessary to carry out some preliminary tests, in order to determine the ohmic resistance of the sensors when subjected to a gas-free environment. It was also realized the capture of the operating curves of the sensors present in the datasheet - graphic elements - that were approximated by polynomials, in order to automate the reading via the Arduino®. The designed apparatus was operationally and functionally efficient in the detection of gases in the environment. The gas concentration measurement, however, can not be adequately characterized, due to the difficulties encountered in creating a reference test environment, that is, it was not possible during the work - for reasons of cost and time - to submit the sensor to a certified environment, where the gas concentration was known with precision and that allowed to confront the known result of the test environment with the information offered by the proposed apparatus. Nevertheless a functional evaluation was made from the construction of a test environment involving a kitchen gas canister, its regulating valve and in a 20 liter container, where the sensor was inserted. The results obtained, the challenges encountered and some contour possibilities are discussed throughout the work. Despite the limitations pointed out in the sensor characterization, the work concludes with the feasibility and potential of the proposed system, where the characteristics of versatility, automation and low cost stand out. The work also points to new studies and developments that can contribute to the production of an innovative product with a promising commercial appeal.

Keywords: Arduino. Gas Sensor. Measuring System.

LISTA DE FIGURAS

Figura 1 -	Disposição dos componentes dos sensores da família MQ-x	13
Figura 2 -	Circuito elétrico dos sensores da família MQ-x	14
Figura 3 -	Circuito divisor de tensão	14
Figura 4 -	Resposta do sensor MQ-4 à diferentes gases	16
Figura 5 -	Influência da temperatura e umidade no sensor MQ-4	16
Figura 6 -	Comportamento do sensor MQ-6 a diversos gases	17
Figura 7 -	Influência da temperatura e umidade no sensor MQ-6	17
Figura 8 -	Comportamento do sensor MQ-8 à diferentes gases	18
Figura 9 -	Influência da temperatura e umidade no sensor MQ-8	19
Figura 10 -	Comportamento do sensor MQ-135 a diversos gases	20
Figura 11 -	Influência da temperatura e umidade no sensor MQ-135	20
Figura 12 -	Sensor DHT22	21
Figura 13 -	Arduino® MEGA 2549	22
Figura 14 -	Circuito do sistema de monitoramento	23
Figura 15 -	Marcação dos pontos correspondente a curva de resposta ao gás GLP do sensor MQ-4	24
Figura 16 -	Comparação entre o gráfico gerado a partir dos pontos e dos gráficos fornecidos pelo fabricante	25
Figura 17 -	Comparação entre os gráficos gerados a partir dos pontos e do gráfico do polinômio	26
Figura 18 -	Interface do sistema	28
Figura 19 -	Fluxograma do software do sistema de monitoramento	29
Figura 20 -	Experimento para teste do sistema de monitoramento de ambiente	30

LISTA DE QUADROS

Quadro 1 -	Sensores e medidores de gases no mercado atual	12
Quadro 2 -	Componentes dos sensores da família MQ-x	13
Quadro 3 -	Informações sobre o sensor DHT22	21
Quadro 4 -	Características do Arduino® MEGA 2549	22
Quadro 5 -	Componentes do projeto	23
Quadro 6 -	Gases que podem ser selecionados para cada sensor MQ-x	28
Quadro 7 -	Valores de densidade para GLP, propano e butano para uma temperatura de 22°C e pressão atmosférica	32
Quadro 8 -	Vazões calculadas para a válvula reguladora de botijão de GLP	32
Quadro 9 -	Valores do experimento para o sensor MQ-4	32
Quadro 10 -	Valores do experimento para o sensor MQ-6	33
Quadro 11 -	Valores de erro médio do sensor MQ-4 e MQ-6	33

LISTA DE TABELAS

Tabela 1 - Resistencia padrão por sensor	31
--	----

LISTA DE SIGLAS

ANP	Agência Nacional do Petróleo, Gás Natural e Biocombustíveis
EES	<i>Engineering Equation Solver</i>
GLP	Gás liquefeito de petróleo
IBP	Instituto Brasileiro de Petróleo, Gás e Biocombustíveis
ppm	Partes por milhão
UM	Umidade relativa

LISTA DE SÍMBOLOS

R_s	Resistencia do elemento resistivo dos sensores da família MQ-x [Ω]
R_L	Resistencia de carga dos sensores da família MQ-x [Ω]
R_O	Resistencia padrão dos sensores da família MQ-x [Ω]
V_C	Tensão de alimentação dos sensores da família MQ-x [V]
V_{RL}	Tensão na resistência de carga dos sensores da família MQ-x [V]
V_H	Tensão de alimentação do aquecedor dos sensores da família MQ-x [V]
Q	Vazão [$\frac{L}{s}$]
ρ	Densidade [$\frac{m^3}{kg}$]
CH_4	Metano
CO	Monóxido de carbono
CO_2	Dióxido de carbono
H_2	Hidrogênio
NH_4	Amônia

SUMÁRIO

1	INTRODUÇÃO	10
2	REFERENCIAL TEÓRICO	12
2.1	Sensor de gás MQ-x	12
2.1.1	Sensor MQ-4	15
2.1.2	Sensor MQ-6	17
2.1.3	Sensor MQ-8	18
2.1.4	Sensor MQ-135	19
2.2	Sensor DHT22	21
2.3	Arduino® MEGA 2549	22
3	PROCEDIMENTO EXPERIMENTAL	23
3.1	Componentes	23
3.2	Montagem	23
3.3	Metodologia de medição	24
3.3.1	Digitalização	24
3.3.2	Calibração	27
3.4	Software	27
3.5	Teste experimental	30
4	ANÁLISE DOS RESULTADOS	31
4.1	Resultados do procedimento de calibração	31
4.2	Resultados do teste experimental	32
5	CONCLUSÃO	34
6	REFERENCIAS	36
	APÊNDICE A - Cálculos dos polinômios do sensor MQ-4	37
	APÊNDICE B - Cálculos dos polinômios do sensor MQ-6	43
	APÊNDICE C - Cálculos dos polinômios do sensor MQ-8	50
	APÊNDICE D - Cálculos dos polinômios do sensor MQ-135	56
	APÊNDICE E - Código do arduino	61
	ANEXO A – Datasheet do sensor MQ-4	90
	ANEXO B – Datasheet do sensor MQ-6	93
	ANEXO C – Datasheet do sensor MQ-8	96
	ANEXO D – Datasheet do sensor MQ-135	99

1 INTRODUÇÃO

Diversos gases são utilizados atualmente, não só na indústria, mas também em residências, para aplicações que vão desde combustível para um simples fogão a complexas máquinas de análises químicas. Muitos desses gases constituem perigo constante à saúde das pessoas e, em alguns casos, até mesmo um perigo letal.

Gases como o gás natural que é composto, em sua maioria, por metano (CH_4), tem se tornado uma das alternativas como fonte de energia mais limpa. Segundo matéria publicada pelo Instituto Brasileiro de Petróleo, Gás e Biocombustíveis, IBP, o Brasil tem estudado maneiras de transferir parte de sua matriz de geração elétrica para uma geração termoeletrica, (IBP, 2017) utilizando o gás natural. Para a utilização do gás natural ele passa por um processo de liquefação, o que facilita o armazenamento e transporte do gás natural, entretanto os tanques de armazenamento e dutos de transporte são passíveis de ruptura ocasionando vazamentos que podem gerar graves riscos devido à própria natureza do gás que é altamente inflamável (RAMOS, 2012). Para evitar esses vazamentos são feitas manutenções periódicas de prevenção, mas raramente existe monitoramento constante.

Outro gás comumente utilizado é o gás liquefeito de petróleo (GLP), que, em sua utilização mais comum, está presente em grande parte das residências brasileiras como combustível para fogões e fornos. Esse gás, também por sua natureza altamente inflamável, apresenta grave risco de incêndio. Além disso, como o gás é mais denso que o ar, em caso de vazamento prolongado em ambiente fechado ou de pouca ventilação, ele se acumula na parte inferior criando um ambiente com baixa concentração de oxigênio, podendo levar a desmaios, asfixia, ou caso entre em contato com uma fonte de ignição pode levar uma explosão violenta (MARCICANO, 2014). Atualmente é utilizado um composto a base de enxofre chamado mercaptana que é adicionado para dar cheiro ao gás que é inodoro e facilitar a percepção de qualquer vazamento do mesmo (Corpo Bombeiros, 2017).

Também temos o monóxido de carbono (CO) que é resultado da combustão não completa de hidrocarbonetos. Um exemplo de fonte de monóxido de carbono muito comum é os gases de escapamento de veículos automotivos que são liberados no ambiente. O monóxido de carbono é um gás inodoro, incolor e altamente tóxico, sendo capaz de deixar uma pessoa inconsciente ou levá-la a óbito rapidamente se exposto a uma concentração muito alta. Isso se deve ao fato que o monóxido de carbono tem uma afinidade muito maior com a hemoglobina, que são responsáveis por transportar oxigênio (LACERDA, 2005).

O hidrogênio (H_2) que vem sendo estudado como possível solução para uma fonte de energia limpa obtida por processos renováveis, como a produção por meio de biomassa, é um gás também inodoro, insípido e incolor, com um elevado calor específico e encontra dificuldades em sua utilização por necessitar ser armazenado ou em baixas temperaturas, cerca de $-235^{\circ}C$, ou em tanques a alta pressão (VARGAS, 2006). Em casos de vazamentos de hidrogênio, esse pode se inflamar ou explodir rapidamente se em contato com alguma fonte de ignição.

Sensores capazes de monitorar esses gases e suas devidas concentrações atualmente variam de R\$ 70,00 a R\$ 4000,00, dependendo da capacidade do mesmo e a precisão de leitura, sendo os sensores mais baratos apenas para detecção, sem a possibilidade de leitura estimada

da concentração e, em geral, para gases mais comuns como o gás liquefeito de petróleo e os mais caros permitem a leitura da concentração de forma precisa e, em alguns casos, de mais de um gás em um mesmo sistema.

Tendo em mente a segurança na utilização desses e outros gases e o preço envolvido para a implementação de sistemas de monitoração dos mesmos, esse trabalho propõem o desenvolvimento de um sistema que seja simples e de baixo custo, porém que seja eficiente e programável para a situação desejada, utilizando da plataforma Arduino® e sensores de gás da família MQ-x fabricados pela empresa Hanwei Sensors e que seja capaz de detectar e monitorar a concentração de um gás, previamente configurado.

Objetivo Geral

Estudar, projetar, construir e testar um sistema que seja programável e de baixo custo para o monitoramento e detecção de gases em um ambiente.

Objetivos Específicos

- a) Projetar um sistema de monitoramento de gases contidos em um ambiente usando a plataforma Arduino® e os sensores de gás da família MQ-x.
- b) Desenvolver um software para o Arduino® capaz de implementar o sistema projetado.
- c) Desenvolver e avaliar uma metodologia de calibração e medição para o sistema.

2 REFERENCIAL TEÓRICO

Atualmente no mercado existem modelos variados de sensores, uns capazes apenas de fazer a detecção do gás e outros capazes medir a concentração de um ou mais gases. O Quadro 1 mostra alguns desses sensores, seus fabricantes e o preço atual de mercado.

Quadro 1- Sensores e medidores de gases no mercado atual

Tipo de sensor	Empresa	Preço (R\$)
Detector de GLP/Gás Natural	Zero Gás	99,00
Detector de GLP/Gás Natural	Suryha	300,00
Detector de gás CD-100 (CH_4 – CO – H_2)	Ridgid	950,00
Medidor de concentração Ventis MX4 (O_2 - CO - H_2S - Ex - Gases Combustíveis)	Industrial Scientific	3950,00

Fonte: Mercado Livre (www.mercadolivre.com, acesso em 22 dez. 2017, 21:30) / Lojas Americanas (www.americanas.com, acesso em 22 dez. 2017, 21:33) / Epibrasil (acesso em 22 dez. 2017, 21:37)/ Ebay (www.ebay.com, acesso em 22 dez. 2017, 21:40)

Existem também no mercado alguns sensores de custo reduzido, mas que necessitam de serem implementados juntamente com algum micro controlador e um circuito apropriado como o Sensor detector de GLP da empresa Homsun Tecnologia que funciona a partir do princípio de combustão catalítica, como descrito em seu datasheet, e custa cerca de R\$ 45,00.

Outro sensor de baixo custo encontrado no mercado é o sensor MC113 para metano, hidrogênio e GLP da empresa Kai Tuo Da que também funciona a partir da combustão catalítica custando R\$ 30,00 e os sensores de gás natural NAP-50A e NAP-55A da empresa Nemoto que possuem o mesmo princípio de funcionamento a partir da combustão catalítica com um custo de R\$ 30,00 a R\$ 60,00 dependendo do modelo.

Além destes sensores acima mencionados encontramos os sensores da família MQ-x da empresa Hanwei Eletronics que, por suas características que são descritas nos itens 2.1 a 2.2, foram escolhidos para o projeto.

2.1 SENSOR DE GÁS MQ-X

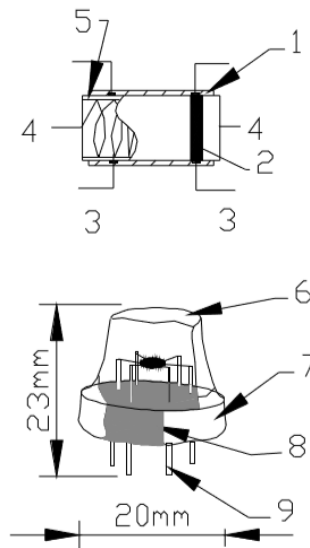
Os sensores de gás da família MQ-x, são sensores de baixo custo, rápidos e de longa vida útil. Segundo o *datasheet* fornecido pelo fabricante, a empresa Hanwei Eletronics, e contidos nos anexos A, B, C e D, estes sensores são compostos por um micro tubo cerâmico com óxido de alumínio, uma camada sensível de Óxido de Estanho e um eletrodo para medida fixado junto a um aquecedor e a uma carcaça de plástico com um trançado feito de aço inoxidável, como pode ser visto na Figura 1 e no Quadro 2.

É importante tomar nota que o aquecedor permite que o sensor esteja em uma temperatura ótima para que a leitura seja precisa e que para que essa temperatura seja estabelecida o fabricante recomenda que o aquecedor fique ligado por no mínimo vinte e quatro horas antes de qualquer medida.

Quadro 2- Componentes dos sensores da família MQ-x

	Partes	Material
1	Camada sensítiva	Óxido de Estanho (SnO_2)
2	Eletrodo	Cobre (Cu)
3	Linhas do eletrodo	Platina (Pt)
4	Bobina do aquecedor	Liga de Níquel-Cromo ($Ni-Cr$)
5	Tubo de cerâmica	Óxido de alumínio (Al_2O_3)
6	Trançado anti explosão	Aço inoxidável
7	Anel de fixação	Cobre e Níquel
8	Base de resina	Baquelite
9	Pinos	Cobre e Níquel

Fonte: *Datasheet* do sensor MQ-4

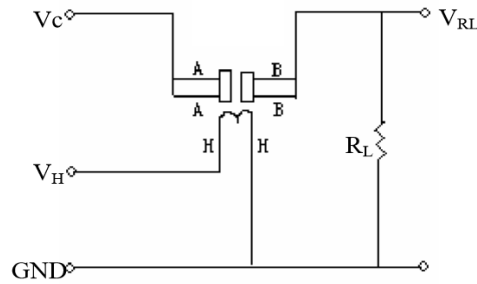
Figura 1 - Disposição dos componentes dos sensores da família MQ-x

Fonte: *Datasheet* do sensor MQ-4

A parte sensítiva tem uma resistência proporcional, de acordo com a característica de cada modelo, à concentração do gás em que o sensor está submetido e se medirmos essa resistência é possível estimar a concentração do gás.

Para realizarmos esta medida utilizamos o circuito elétrico mostrado na Figura 2, retirada do *datasheet* fornecido pelo fabricante, onde V_C é tensão de alimentação do sensor, V_H é a tensão de alimentação para o aquecedor, R_L é uma resistência de carga e V_{RL} é a tensão de na resistência de carga.

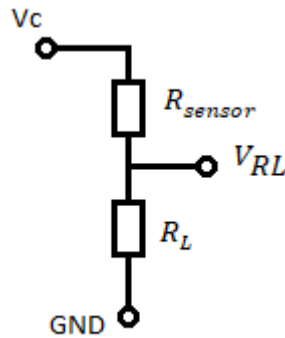
Figura 2 - Circuito elétrico dos sensores da família MQ-x



Fonte: Datasheet do sensor MQ-4

Como foi dito anteriormente, a junção A-B, ou seja, a parte sensível é um elemento resistivo, assim, ao analisar o circuito, pode-se ver que se trata de um circuito divisor de tensão como mostrado na Figura 3 onde a saída será proporcional à entrada e a relação entre as duas resistências.

Figura 3 - Circuito divisor de tensão



Fonte: Autoria Própria

Observando o circuito e aplicando a Lei de Ohm temos que:

$$V_C = (R_L + R_{sensor}) * i \quad (1)$$

$$V_{RL} = R_L * i \quad (2)$$

Como a corrente i é a mesma nas duas equações, pode-se juntar (1) e (2) e obter como resultado a equação (3):

$$V_{RL} = \frac{R_L}{R_L + R_{sensor}} \cdot V_C \quad (3)$$

Desta relação controlamos a tensão de entrada, o resistor de carga e podemos medir a tensão no resistor de carga permitindo assim calcular a resistência do sensor e a partir dela inferir a concentração de gás, logo a formula anterior pode ser rearranjada para a forma da equação (4):

$$R_{sensor} = \left(\frac{V_C}{V_{RL}} - 1 \right) \cdot R_L \quad (4)$$

Os vários modelos de sensores da família MQ-x possuem mesma disposição de elementos assim como mesmo princípio de funcionamento e utilizam do mesmo circuito descrito anteriormente e por isso, no geral, são influenciados de alguma forma por toda a gama de gases dos quais o sensor consegue medir, podendo gerar assim ruídos nas medições caso mais de um desses gases estejam presentes no ambiente monitorado. Assim torna-se necessário a avaliação de quando cada sensor pode ser utilizado ou não para evitar gerar leituras errôneas devido a interferências não previamente consideradas.

Outra questão importante é o que todos os sensores da família MQ-x sofrem influência da temperatura e umidade em suas leituras. Para contornar este problema foi utilizado um sensor de temperatura e umidade DHT22 que é discutido no item 2.2.

Visto as características dos sensores MQ-x, foram escolhidos para este projeto quatro sensores: MQ-4, MQ-6, MQ-8 e MQ-135.

2.1.1 Sensor MQ-4

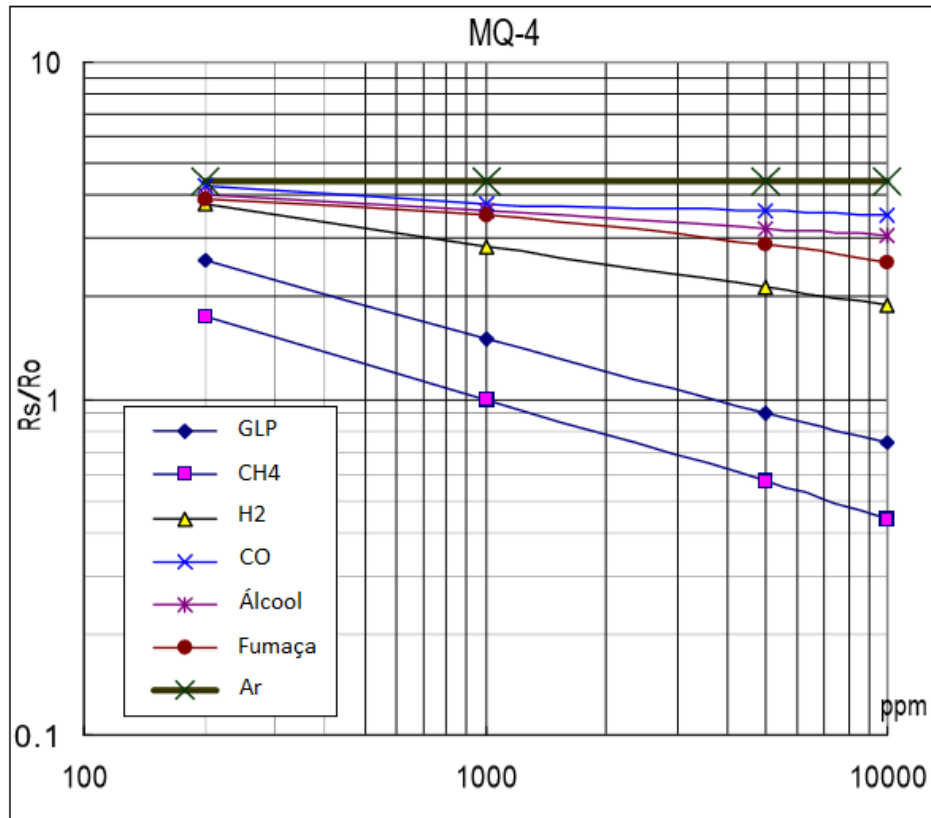
O sensor MQ-4 tem uma boa sensibilidade a gases combustíveis como butano, propano e metano. Na Figura 4, retirada do I disponibilizado pelo fabricante, podemos ver como o sensor reage a diferentes gases.

Observando a Figura 4, podemos ver que o gráfico da resposta do sensor aos diferentes gases fica em função de $\frac{R_s}{R_o}$, onde R_s é a resistência do sensor a determinada concentração de gás e R_o a resistência do sensor em um ambiente com uma concentração de 1000 partes por milhão (ppm) de metano, e da concentração em ppm do gás no ambiente medido.

Podemos notar também que o sensor apresenta alta resistência quando o ar está limpo e uma boa sensibilidade, ou seja, alta variação de resistência em um ambiente contendo hidrogênio, metano e gás liquefeito de petróleo. Visto isso concluímos que este sensor é melhor utilizado para medir metano ou GLP evitando a influência de hidrogênio, álcool e propano e o sensor possui um limite de medida entre 200 a 10000 ppm.

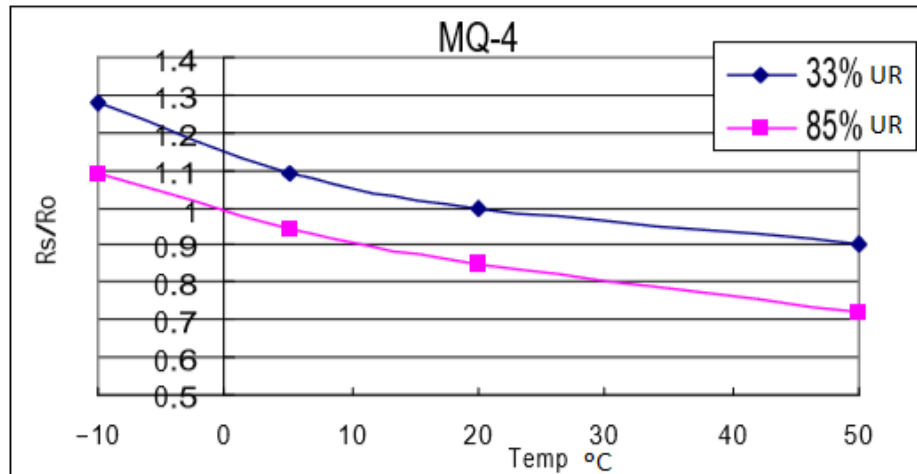
Como dito anteriormente os sensores da família MQ-x são influenciados diretamente pela temperatura e umidade e por isso necessita de correções em suas leituras proporcionais à temperatura e umidade no qual se encontram. Segue a Figura 5, retirada do *datasheet* fornecido pelo fabricante, onde podemos ver a influência da temperatura e umidade no sensor MQ-4.

Figura 4 - Resposta do sensor MQ-4 à diferentes gases



Fonte: Adaptado do *Datasheet* do sensor MQ-4

Figura 5 - Influência da temperatura e umidade no sensor MQ-4



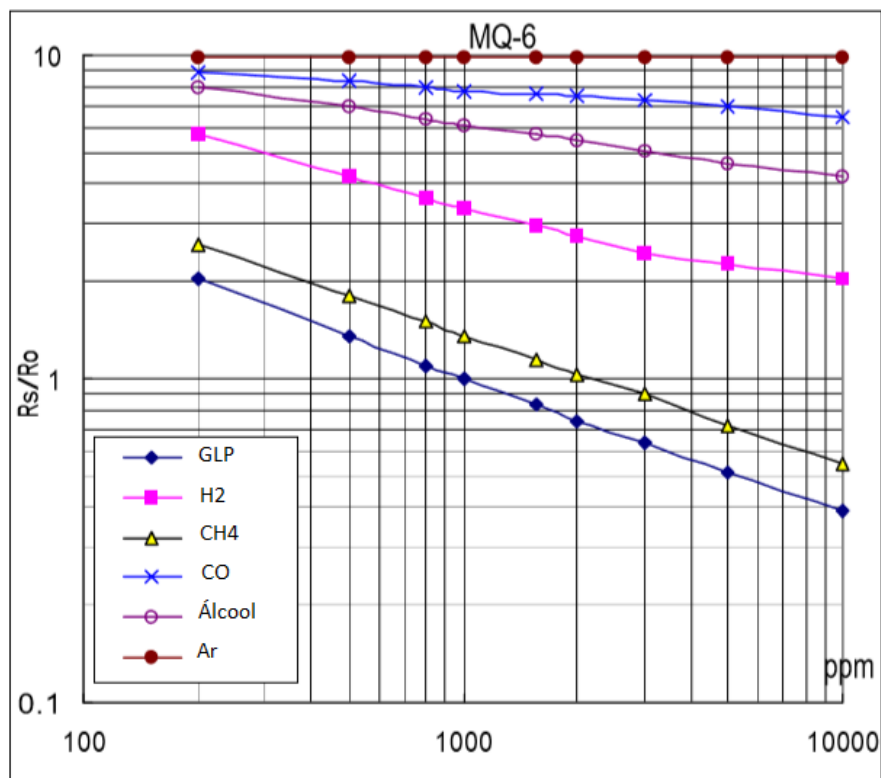
Fonte: Adaptado do *Datasheet* do sensor MQ-4

2.1.2 Sensor MQ-6

O sensor MQ-6 pode ser utilizado para medir alguns gases combustíveis como hidrogênio e GLP.

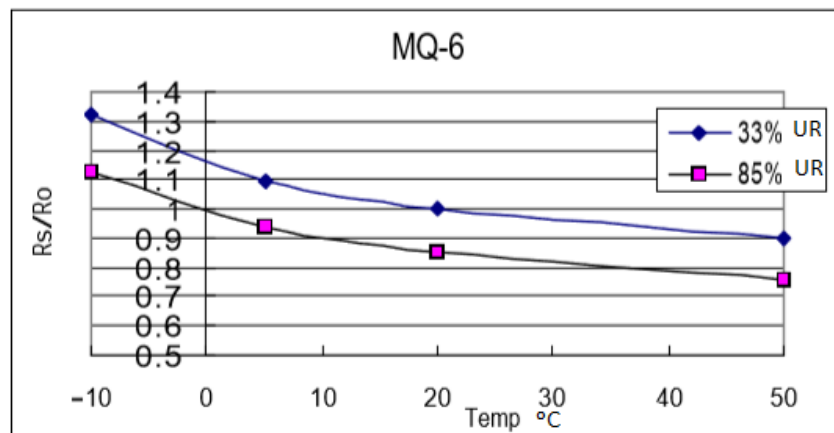
A Figura 6, retirada do *datasheet* fornecido pelo fabricante, mostra o comportamento do sensor em contato com diversos gases e a Figura 7, também retirada do *datasheet* do fabricante, mostra a influência da temperatura e umidade no sensor MQ-6.

Figura 6 - Comportamento do sensor MQ-6 a diversos gases



Fonte: Adaptado do *Datasheet* do sensor MQ-6

Figura 7 - Influência da temperatura e umidade no sensor MQ-6



Fonte: Adaptado do *Datasheet* do sensor MQ-6

Como podemos observar na Figura 6, é relacionado novamente a relação $\frac{R_s}{R_o}$ e a concentração em ppm do gás medido, onde R_o é, neste caso, a resistência do sensor a um ambiente com 1000 ppm de GLP. Podemos ver como o sensor tem baixíssima condutividade quando exposto ao ar limpo e baixa sensibilidade a álcool e ao monóxido de carbono, além de uma excelente sensibilidade ao GLP e ao metano.

Assim podemos concluir que sua aplicabilidade mais adequada se dá em situações onde desejamos evitar a influência de álcool e fumaças como fumaça de cozinha ou de cigarros e desejamos medir GLP ou metano, respeitando um limite de medição entre 200 a 10000 ppm.

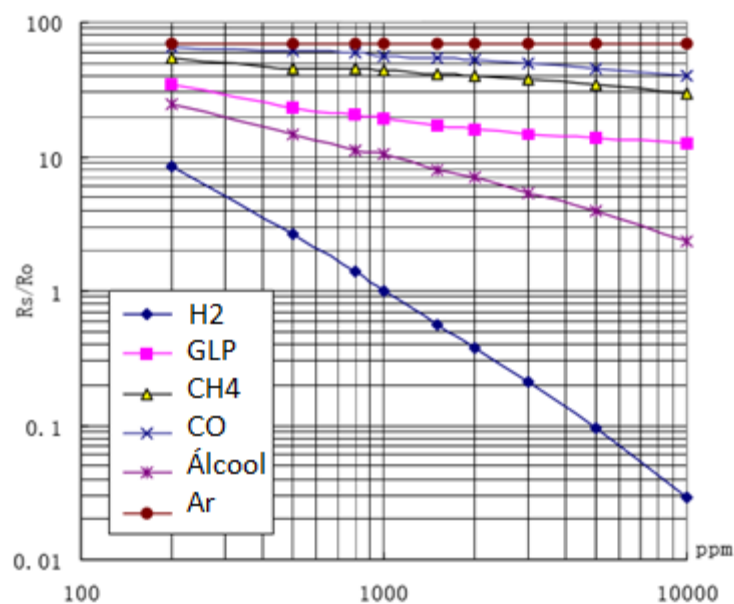
2.1.3 Sensor MQ-8

O sensor MQ-8 pode ser utilizado para medir hidrogênio, álcool, monóxido de carbono e GLP.

As Figuras 8 e 9, retiradas do *datasheet* fornecido pelo fabricante, mostram as respectivamente o comportamento do sensor MQ-8 a diversos gases e a influência da temperatura e umidade em suas medidas.

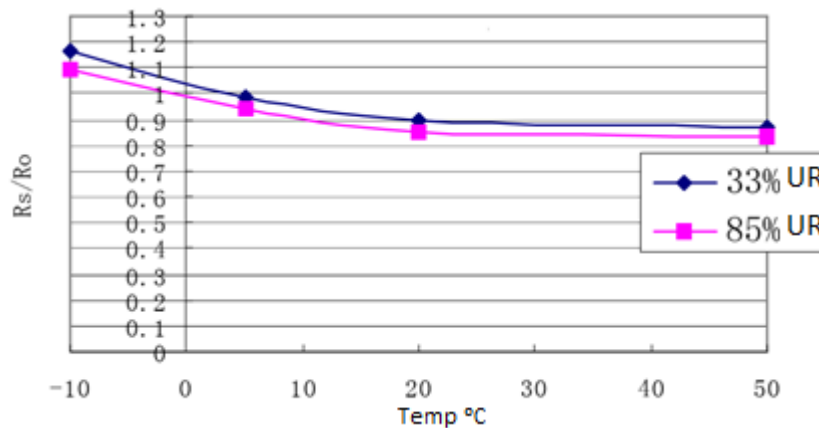
Novamente, observando a Figura 8, vemos que o comportamento do sensor é relacionado com a razão $\frac{R_s}{R_o}$ e a concentração em ppm do gás medido, onde, aqui, R_o é a resistência do sensor para uma concentração de 1000 ppm de hidrogênio. Notamos também que o sensor possui uma sensibilidade muito alta para o hidrogênio e, ao mesmo tempo, muito baixa para os demais gases e o limite de medição varia entre 200 a 10000 ppm. Assim podemos concluir que esse sensor é mais adequado para medição do hidrogênio, evitando o ruído de gases como GLP, álcool, metano e fumaças.

Figura 8 - Comportamento do sensor MQ-8 à diferentes gases.



Fonte: Adaptado do *Datasheet* do sensor MQ-8

Figura 9 - Influência da temperatura e umidade no sensor MQ-8.



Fonte: Adaptado do *Datasheet* do sensor MQ-8

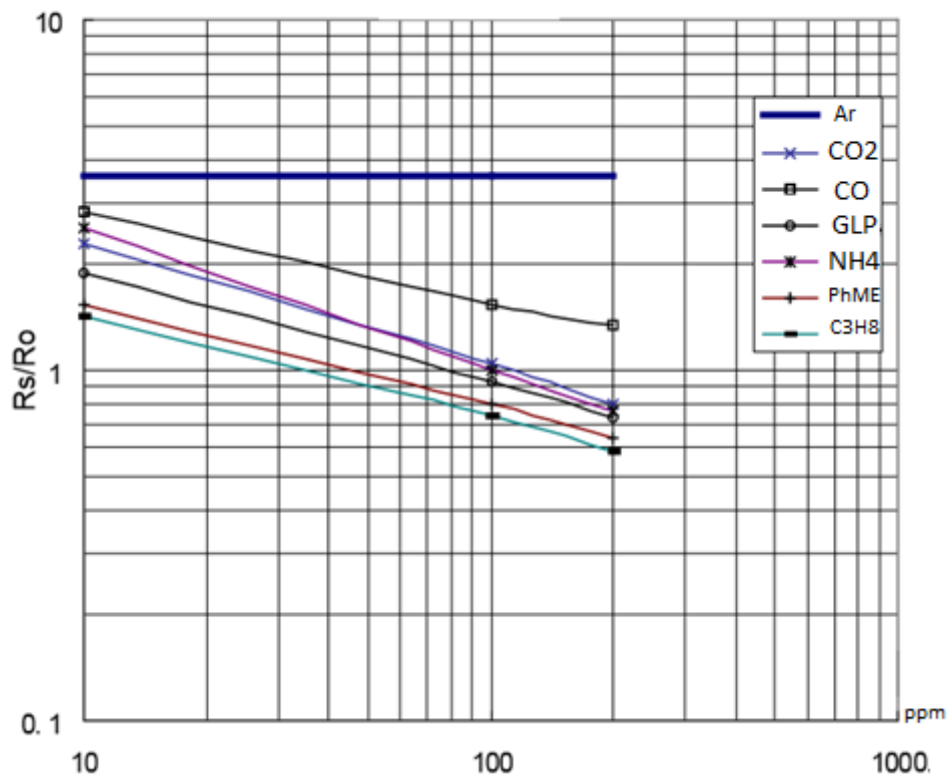
2.1.4 Sensor MQ-135

Por último temos o sensor MQ-135. Este sensor tem a capacidade de medir amônia, benzeno, fumaças, gás carbônico, dentre outros.

Podemos observar o comportamento do sensor MQ-135 em diversos gases na Figura 10 e a influência da temperatura e umidade na Figura 11, ambas tiradas do *datasheet* fornecido pelo fabricante.

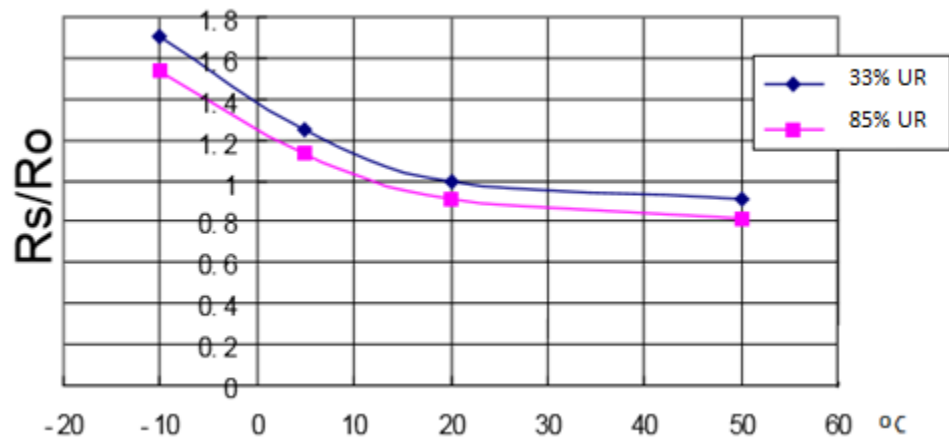
Como os demais sensores vemos que o comportamento do sensor MQ-135 é avaliado usando a relação da razão $\frac{R_s}{R_o}$ e a concentração em ppm do gás medido. R_o é a resistência do sensor a uma concentração de 100 ppm de amônia. Além disso, podemos ver que o sensor possui uma sensibilidade muito próxima para todos os gases dos quais ele permite medir e isso nos leva concluir que este sensor não pode ser usado em ambientes onde haverá mais de um desses gases, pois suas leituras possuirão muito ruído e, por isso, não serão leituras confiáveis. É possível observar também que o limite de medição é bem baixo, variando entre 10 e 200 ppm.

Figura 10 - Comportamento do sensor MQ-135 a diversos gases



Fonte: Adaptado do *Datasheet* do sensor MQ-135

Figura 11 - Influência da temperatura e umidade no sensor MQ-135.



Fonte: Adaptado do *Datasheet* do sensor MQ-135.

2.2 Sensor DHT22

Como mostrado anteriormente na descrição dos sensores da família MQ-x, esses sensores são influenciados diretamente pela temperatura e pela umidade do ambiente. Pensando nisso foi escolhido o sensor DHT22 para realizar a medição da temperatura e umidade.

O sensor DHT22 é um sensor de baixo custo e uso muito simplificado, capaz de realizar as duas medidas complementares do projeto, a medida de temperatura e a medida de umidade, com uma precisão de 0,5 °C e 2% UR respectivamente e um tempo de resposta de 2 segundos.

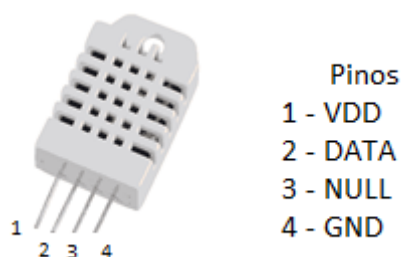
Os dados do sensor podem ser encontrados no Quadro 3 e as conexões dos pinos do sensor na Figura 12.

Quadro 3 - Informações sobre o sensor DHT22

Tensão de entrada	3,3-6 V DC
Tipo de saída	Sinal digital
Alcance de operação para umidade	0 – 100 %UR
Alcance de operação para temperatura	-40 – 80 °C
Precisão para umidade	+/- 2% UR
Precisão para temperatura	+/- 0,5 °C
Resolução para umidade	0,1 %UR
Resolução para temperatura	0,1 °C
Tempo de medição	2 s

Fonte: Datasheet do sensor DHT22

Figura 12 - Sensor DHT22



Fonte: Adaptado do site Arduino e Cia (www.arduinoecia.com.br)

O sensor MQ-x utiliza uma porta analógica para a leitura da voltagem gerada pelo circuito divisor de tensão discutido anteriormente. Já o sensor DHT22 utiliza de uma mesma porta digital para a leitura tanto da temperatura quanto da umidade.

2.3 Arduino® MEGA 2549

A plataforma Arduino® é uma plataforma de desenvolvimento barata e de fácil acesso. Permite uma excelente customização para diferentes aplicações, visto que possui diversos modelos com processadores variados, diferentes números de entradas e saídas digitais, e analógicas, entre outras funções. Além disto, já possui um amplo grupo de desenvolvedores que trabalham com a plataforma criando dos mais diversos projetos.

Dentre os modelos do Arduino® o escolhido foi o Arduino® MEGA 2549. Este modelo é similar à versão básica da plataforma Arduino®, o Arduino® UNO, com mesmo processamento porem com mais portas digitais e analógicas.

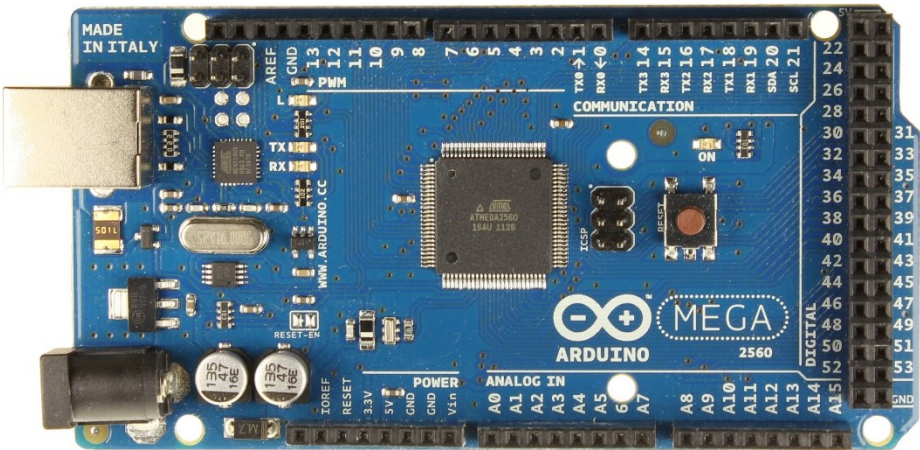
As características do Arduino® MEGA 2549 podem ser vistas no Quadro 4 e Figura 13.

Quadro 4 - Características do Arduino® MEGA 2549

Microprocessador	ATmega2560
Voltagem de operação	5 V
Voltagem de entrada	7-12 V
Pinos de entrada/saída digital	54 (15 possuem saída PWM)
Pinos de entrada analógica	16
Corrente DC por pino digital	20 mA
Corrente por pino de 3.3V	50 mA
Clock	16 MHz

Fonte: Site Arduino (Arduino,2017, www.arduino.cc)

Figura 13 - Arduino® MEGA 2549



Fonte: Site Arduino (Arduino, 2017, www.arduino.cc)

3 PROCEDIMENTO EXPERIMENTAL

3.1 Componentes

Para que o sistema de monitoramento cumprisse a proposta de um sistema de baixo custo e programável foi selecionado a plataforma Arduino® e o conjunto de sensores da família MQ-x, mais especificamente os componentes listados no Quadro 5 que, no total, têm um preço de R\$ 80,00.

Quadro 5 - Componentes do projeto

Quantidade	Componente	Preço (R\$)
1	Arduino® MEGA 2549	30,00
1	Sensor de Gás MQ-6	7,00
1	Sensor de Gás MQ-8	7,00
1	Sensor de Gás MQ-4	7,00
1	Sensor de Gás MQ-135	10,00
1	Sensor de temperatura e humidade DHT22	12,00
1	Shield LCD para Arduino	7,00

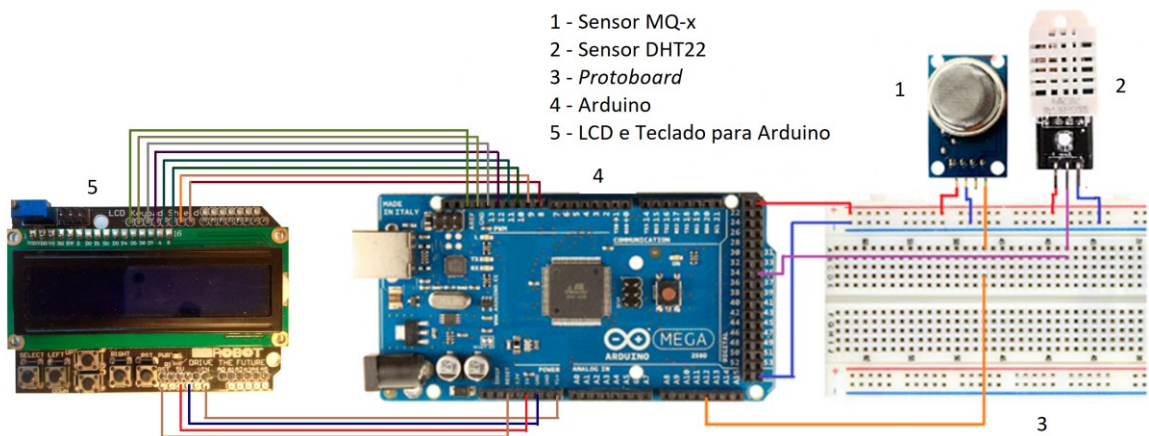
Fonte: Autoria própria

3.2 Montagem

Para a montagem do projeto primeiro decidimos qual sensor da família MQ-x será usado de acordo com a necessidade da aplicação

A Figura 14 mostra a montagem realizada para o projeto.

Figura 14 - Circuito do sistema de monitoramento



Fonte: Autoria própria

Segundo o *datasheet* dos sensores MQ-x e do DHT22 utilizaremos a saída de 5 V do Arduino® para alimentação dos mesmos.

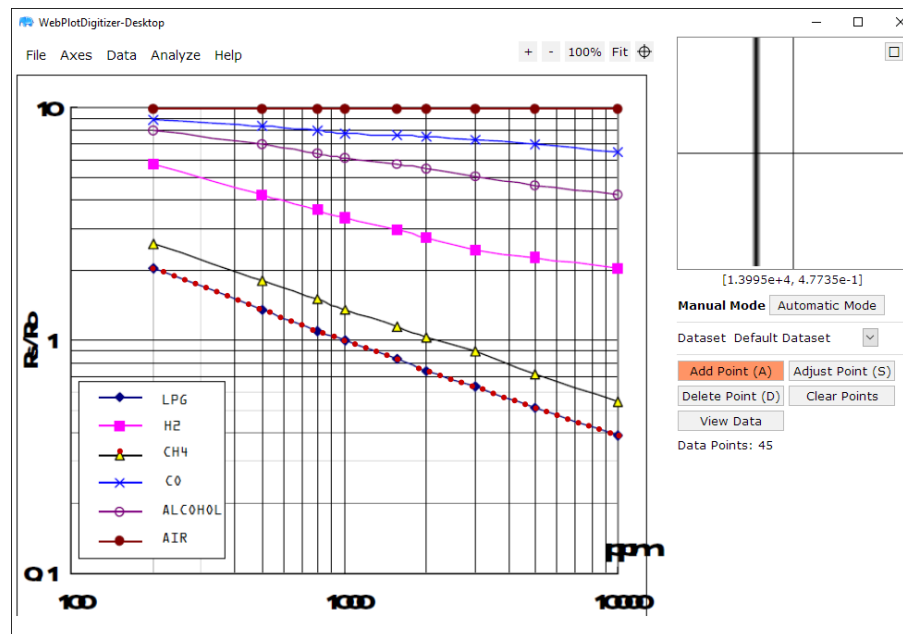
3.3 Metodologia de medição

Para realizar as medições utilizaremos dos gráficos fornecidos pelo fabricante para relacionar o valor da resistência do sensor medida a uma concentração de um gás específico e para podermos utilizar destes gráficos devemos considerar sempre uma concentração fixa de oxigênio no ambiente de 21%. Assim o primeiro passo é encontrar polinômios que representem as curvas dos gráficos tanto do comportamento do sensor quando o da influência da temperatura e umidade.

3.3.1 Digitalização

Utilizando do software gratuito chamado *WebPlotDigitizer* marcamos nos gráficos os pontos correspondentes à suas curvas manualmente e a ferramenta retorna uma tabela contendo as coordenadas de cada um desses pontos marcados levando em consideração a escala logarítmica como mostra a Figura 15.

Figura 15 - Marcação dos pontos correspondente a curva de resposta ao gás GLP do sensor MQ-4

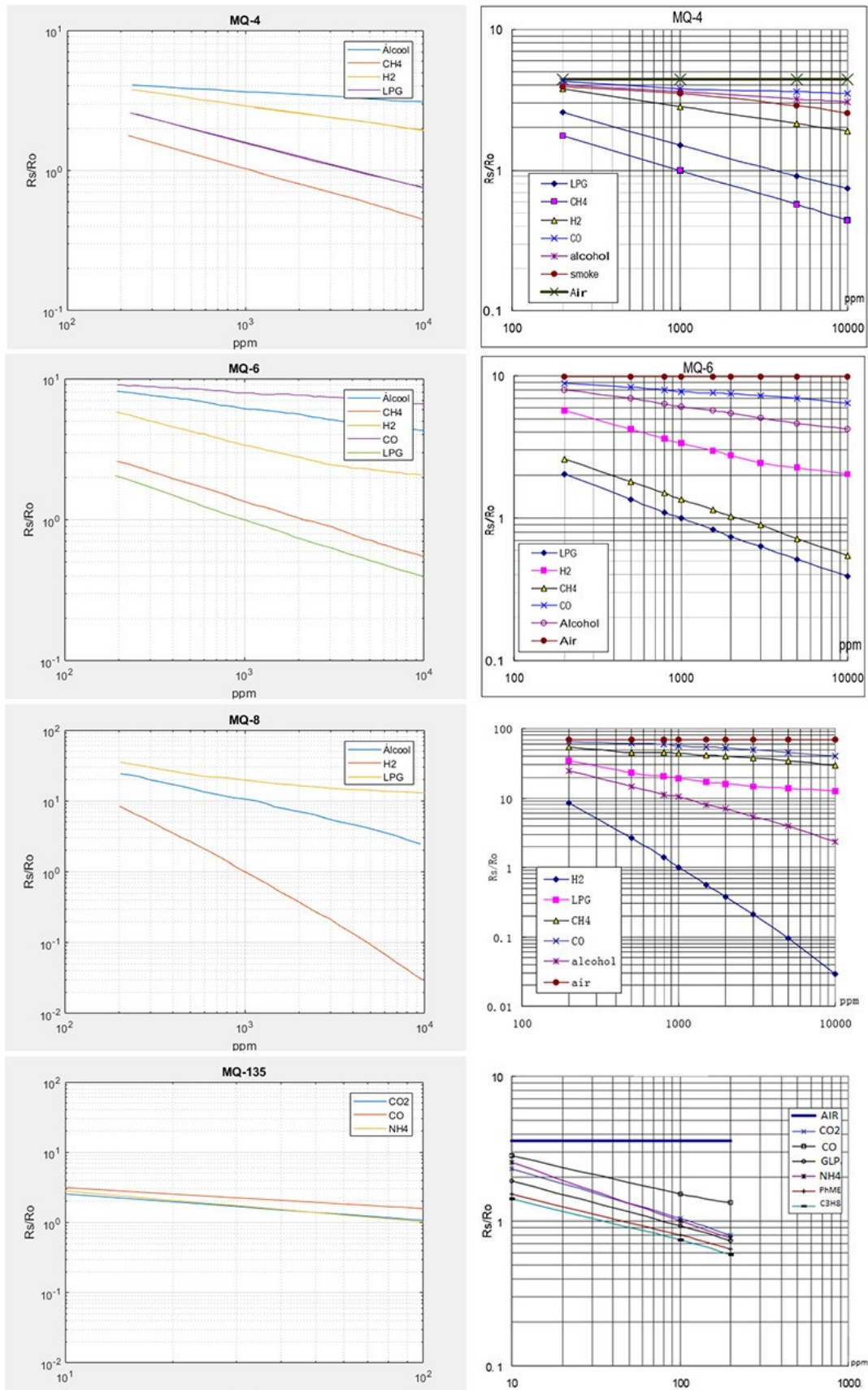


Fonte: Autoria própria

Esse procedimento é repetido para cada curva de gás de cada sensor, assim como as curvas de seus respectivos gráficos da influência da temperatura e umidade.

De posse dos pontos de todos os gráficos, os pontos são transferidos para o software MATLAB® e são plotados para comparação como mostra a Figura 16.

Figura 16 - Comparação entre o gráfico gerado a partir dos pontos e dos gráficos fornecidos pelo fabricante.



Fonte: Autoria própria

Como os gráficos são logarítmicos, transformamos os dados usando na equação (5).

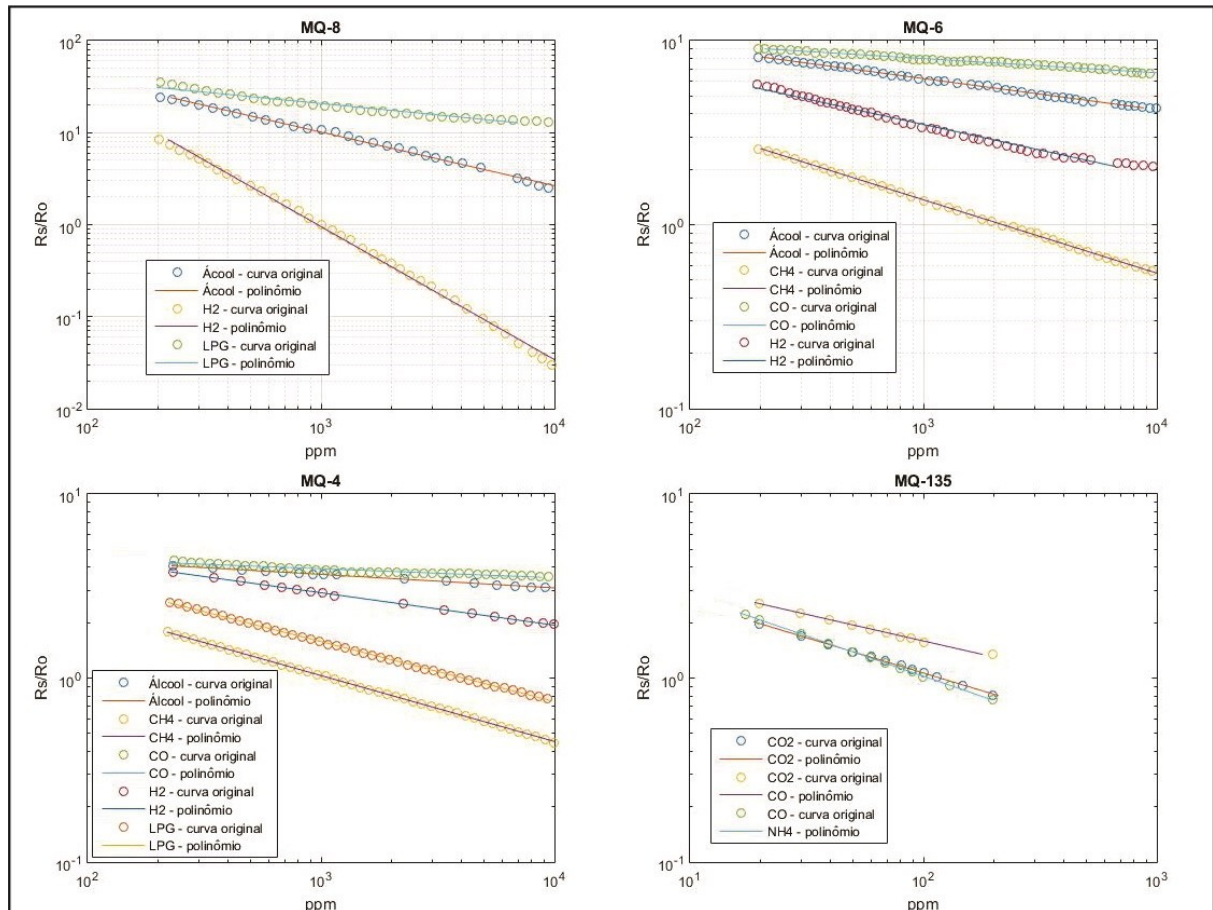
$$Y = \log(x) \quad (5)$$

Onde x são os dados obtidos por meio do software e Y são os dados transformados. Em seguida utilizamos da função polyfit, a qual cria um polinômio cujo o gráfico se aproxima dos pontos desejados e a função polyval, que avalia o polinômio para podermos comparar as curvas encontradas com os pontos iniciais.

Esse procedimento foi realizado para todos os gráficos e o código utilizado para implementar esse procedimento pode ser encontrado nos apêndices 1 a 4.

A Figura 17 mostra o gráfico dos polinômios encontrados comparado aos gráficos gerado a partir da tabela de pontos correspondente ao comportamento do sensor.

Figura 17 - Comparação entre os gráficos gerados a partir dos pontos e do gráfico do polinômio



Fonte: Autoria própria

3.3.2 Calibração

Para que possamos utilizar dos gráficos fornecidos pelo fabricante e assim os polinômios encontrados a partir destes é necessário calcular a razão $\frac{R_s}{R_o}$. Como discutido anteriormente R_s pode ser medido, porem R_o depende de um ambiente controlado com uma concentração específica de um gás para cada modelo de sensor.

Uma opção para calcular o R_o seria novamente recorrer ao gráfico. Podemos notar, se observarmos o gráfico de qualquer um dos sensores, Figura 4, 6, 8 ou 10, todos eles mostram a resposta do sensor ao ar limpo e, em todos, podemos notar que essa resposta é um valor constante. Assim, para encontrar R_o , podemos levar o sensor a um local com o ambiente considerado limpo, como um ambiente aberto e arejado, e fazer a medida da resistência do sensor nesse local e com o valor da constante retirado do gráfico do sensor em questão pode ser calculado facilmente o valor de R_o , o qual será utilizado para fazer todas as outras medidas.

$$\frac{R_s}{R_o} = cte \quad (6)$$

3.4 Software

O *software* implementado para o sistema é dividido em duas partes, configuração e leitura.

A parte de configuração segue os seguintes passos:

1 - Inicialmente é apresentado ao usuário uma interface onde pode ser selecionado o modelo do sensor dentre as opções pré-programadas: MQ-4, MQ-6, MQ-8 e MQ-135.

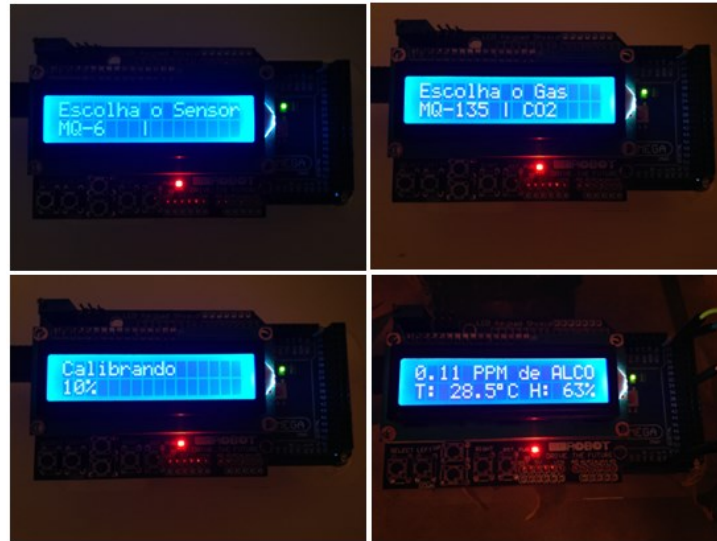
2 - Uma vez selecionado é apresentado a gama de gases disponíveis para o sensor selecionado que podem ser vistos no Quadro 6.

3 - O usuário então tem a opção de deixar o sensor aquecer antes da leitura mostrando apenas um *timer* com o tempo restante para um aquecimento adequado, mas que pode ser cancelado a qualquer momento que o usuário queira.

4 - Por fim é dada a opção de utilizar um valor de calibração padrão ou realizar uma nova calibração. É importante lembrar que para uma nova calibração é necessário que o sistema esteja em um ambiente arejado e livre de qualquer gás que possa afetar o sensor a ser calibrado.

A Figura 18 mostra a interface com o usuário em diversas etapas do *software*.

Figura 18 - Interface do sistema.



Fonte: Autoria própria

Quadro 6 - Gases que podem ser selecionados para cada sensor MQ-x

MQ-4	MQ-6	MQ-8	MQ-135
Álcool	Álcool	Álcool	CO
CH4	CH4	CH4	CO2
CO	CO	H2	NH4
H2	H2	GLP	-
GLP	GLP	-	-

Fonte: Autoria própria

Para a segunda parte, a de leitura, o *software* entra em um *loop*, que finaliza apenas com o desligamento ou reinicialização do sistema, realizando as seguintes tarefas:

1 - Primeiro ele faz uma leitura do sensor MQ-x selecionado obtendo uma relação $\frac{R_s}{R_o}$

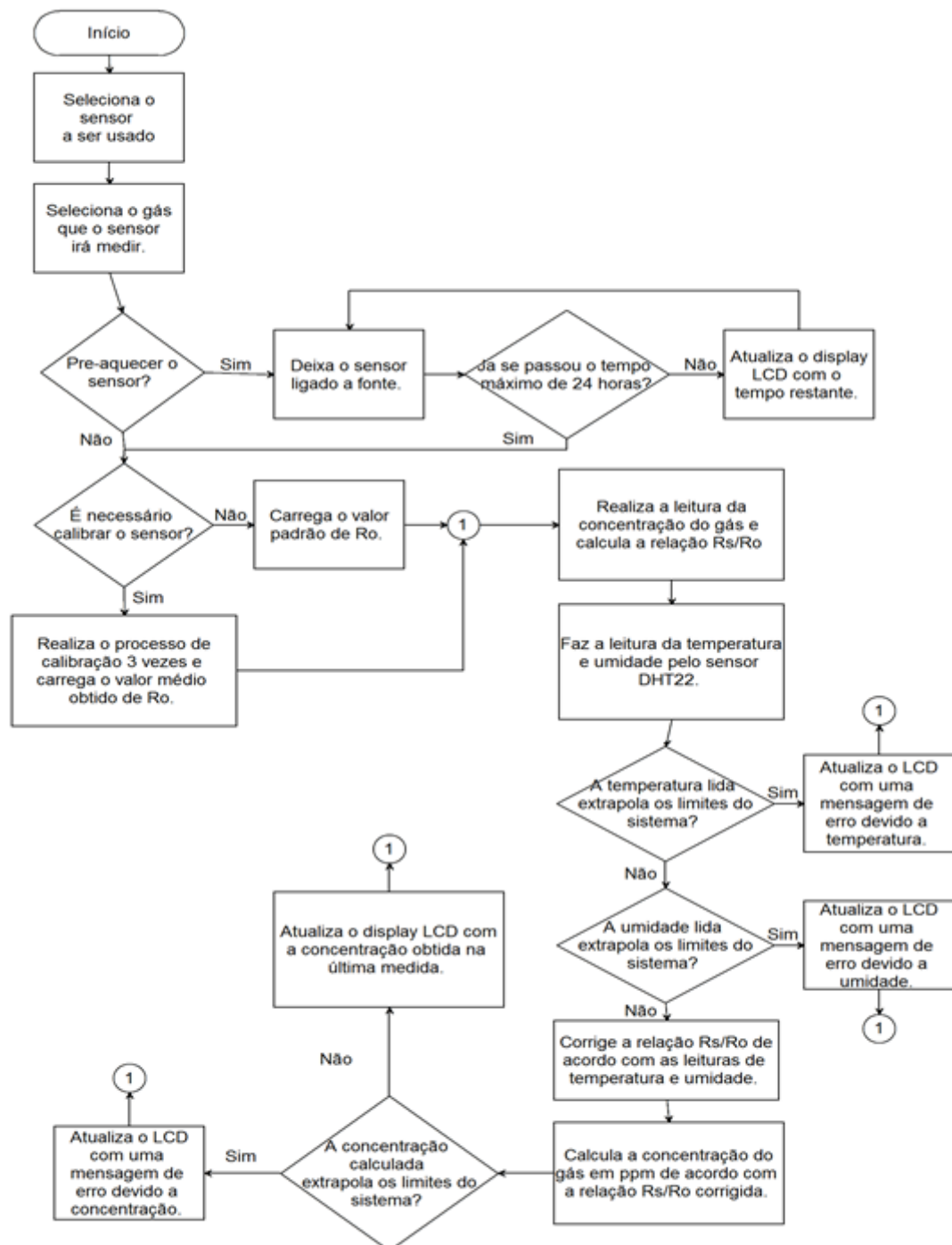
2 - Em seguida realiza a leitura do sensor DHT22 obtendo os valores de temperatura e umidade e os compara com os limites aceitáveis do sensor escolhido previamente, caso estejam dentro desse limite o *software* usa esses valores como entrada no polinômio que representa a curva de influência de temperatura obtendo um fator de correção.

3 - O fator de correção é então aplicado à relação obtida anteriormente do sensor MQ-x e então usa o resultado como entrada no polinômio que representa a curva de resposta do sensor em relação à concentração em ppm do gás que está sendo medido. Caso o valor obtido esteja dentro os limites de leitura do sensor o valor é mostrado no display.

Para as leituras do sensor DHT22 é utilizado a biblioteca DHTlib criada por Rob Tillaart e disponibilizada gratuitamente em seu site. Essa biblioteca fornece funções prontas para a leitura de temperatura e umidade com sensores da família DHT.

Todo o *software* implementado para realizar o procedimento descrito acima pode ser encontrado comentado no apêndice E e seu fluxograma na Figura 19.

Figura 19 - Fluxograma do *software* do sistema de monitoramento



Fonte: Autoria própria

3.5 Teste experimental

Para avaliar o sistema experimentalmente é necessário um ambiente que contenha uma concentração conhecida de um gás para compararmos com o valor da concentração medida pelo sistema neste mesmo ambiente.

De posse de um vasilhame com um volume conhecido de 20,3 litros, foi inserido o sensor de temperatura e um dos sensores capaz de medir GLP. Em seguida foi conectado com uma mangueira esse vasilhame à válvula reguladora de um botijão de GLP e o vasilhame foi selado com silicone para obtermos assim um ambiente com a concentração de GLP controlada.

O controle da quantidade de GLP inserida no vasilhame foi feita por meio da vazão mássica de GLP fornecida pela válvula reguladora que é de 1 kg/h. Como a concentração medida pelo sensor é uma relação entre os volumes, utilizamos da densidade para transformar a vazão mássica em vazão volumétrica como mostra a equação a (7).

$$Q = \frac{m}{t} \left[\frac{kg}{h} \right] * \frac{V}{m} \left[\frac{m^3}{kg} \right] = \frac{V}{t} \left[\frac{m^3}{h} \right] \quad (7)$$

Segundo a Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP), o GLP é uma mistura de propano e butano com proporções variadas, dependendo do local de fornecimento (ANP, 2017), sua densidade considerada foi uma média ponderada da densidade do propano e butano na temperatura medida no momento do experimento. Com o *software Engineering Equation Solver* (EES) foi encontrado o valor desta densidade média, em seguida o valor da vazão volumétrica e por fim foi convertida de $\frac{m^3}{h}$ para $\frac{litros}{s}$.

Com o auxílio de um cronômetro foi possível estimar o volume de GLP inserido em um vasilhame medindo o tempo em que a válvula reguladora fica aberta com uma vazão conhecida. Como o volume do vasilhame é fixo podemos calcular a razão de volumes, o que resulta na concentração de GLP permitindo uma comparação com o valor medido pelo sistema. Neste procedimento o aumento de pressão do vasilhame foi desconsiderado.

O experimento montado pode ser visto na Figura 20.

Figura 20 - Experimento para teste do sistema de monitoramento de ambiente



- 1 - Vasilhame de 20,3 L
- 2 - Sistema de monitoramento
- 3 - Válvula reguladora de vazão

Fonte: Autoria própria

4 ANÁLISE DOS RESULTADOS

4.1 RESULTADOS DO PROCEDIMENTO DE CALIBRAÇÃO

Para o procedimento de calibração, o sistema foi colocado no modo de aquecimento do sensor e, em seguida, foi transferido para um ambiente aberto e bem arejado. Neste local foi realizada a calibração por 10 vezes para que no final chegássemos a um resultado padrão para ser implementado no código. O procedimento foi realizado para cada sensor e os resultados obtidos são mostrados na Tabela 1.

Tabela 1 - Resistência padrão por sensor

Sensor	MQ-4	MQ-6	MQ-8	MQ-135
$R_{o1} (\Omega)$	21750	31500	17400	27000
$R_{o2} (\Omega)$	23750	33250	19000	28500
$R_{o3} (\Omega)$	23500	30800	18800	26400
$R_{o4} (\Omega)$	21250	32200	17000	27600
$R_{o5} (\Omega)$	22500	29750	18000	25500
$R_{o6} (\Omega)$	23000	32200	18400	27600
$R_{o7} (\Omega)$	22000	35000	17600	30000
$R_{o8} (\Omega)$	21250	32900	17000	28200
$R_{o9} (\Omega)$	22250	30100	17800	25800
$R_{o10} (\Omega)$	22750	32200	18200	27600
Média	22375	31893,75	17900	27337,5
Desvio padrão	728,20	981,21	538,52	841,04
Coeficiente de Variação	0,03	0,03	0,03	0,03

Fonte: Autoria própria

Os valores da média e desvio padrão foram calculado descartando o maior e o menor valor de cada sensor que estão destacados em negrito na Tabela 1 e o resultado foi utilizado como padrão de R_o no *software* implementado.

4.2 RESULTADOS DO TESTE EXPERIMENTAL

A densidade média do GLP, a densidade do propano e butano são vistos no Quadro 7 considerando uma proporção de 50% de propano e 50% de butano na mistura, pressão atmosférica e uma temperatura de 22 °C que foi obtida no momento do experimento.

Quadro 7 - Valores de densidade para GLP, propano e butano para uma temperatura de 22°C e pressão atmosférica.

ρ Propano ($\frac{m^3}{kg}$)	ρ Butano ($\frac{m^3}{kg}$)	ρ GLP ($\frac{m^3}{kg}$)
1,812	2,333	2,072

Fonte: Tabelas termodinâmicas do *software* EES

Os valores da vazão mássica e volumétrica podem ser vistos no Quadro 8.

Quadro 8 - Vazões calculadas para a válvula reguladora de botijão de GLP

Vazão mássica ($\frac{kg}{h}$)	Vazão volumétrica ($\frac{m^3}{h}$)	Vazão volumétrica ($\frac{litros}{s}$)
1	0,4825	0,134

Fonte: Autoria própria

O teste foi realizado depois do aquecimento prévio do sensor MQ-4 e MQ-6. Cada sensor foi exposto a um ambiente fechado com cinco concentrações diferentes e o vasilhame foi ventilado entre cada medida.

Como o controle da válvula reguladora foi feito manualmente se tornou inviável controlar a vazão de GLP para o vasilhame de modo que fosse obtido mais que 5 concentrações diferentes, pois o tempo máximo que a válvula poderia ficar aberta é de 1,5 segundo para que a concentração não superasse o limite de medição do sensor, o que gera uma janela muito pequena para um controle manual da válvula.

Os resultados obtidos podem ser encontrados nos Quadros 9 e 10.

Quadro 9 - Valores do experimento para o sensor MQ-4

Temp. (°C)	Humidade (%)	Tempo (s)	Volume de GLP (litros)	Concentração estimada (ppm)	Concentração medida (ppm)	Erro (ppm)	Erro (%)
28	73	0,4	0,0536	2640	2012	628	23,8
28	73	0,6	0,0804	3961	3129	832	21,0
28	73	0,9	0,1206	5941	5054	887	14,9
28	73	1,1	0,1474	7261	6400	861	11,9
28	73	1,3	0,1742	8581	7656	925	10,8
28	73	1,5	0,2144	10562	Limite 10000	-	-

Fonte: Autoria própria

Quadro 10 - Valores do experimento para o sensor MQ-6

Temp. (°C)	Humidade (%)	Tempo (s)	Volume de GLP (litros)	Concentração estimada (ppm)	Concentração medida (ppm)	Erro (ppm)	Erro (%)
28	73	0,3	0,0402	1980	1448	532	26,9
28	73	0,6	0,0804	3961	3194	767	19,4
28	73	0,9	0,1206	5941	5107	834	14,0
28	73	1,2	0,1608	7921	7070	851	10,7
28	73	1,4	0,1876	9380	8417	963	10,3
28	73	1,6	0,2144	10562	Limite 100000	-	-

Fonte: Autoria própria

O Quadro 11 mostra os valores do erro médio, desvio padrão e coeficiente de variação encontrados.

Quadro 11 - Valores de erro médio do sensor MQ-4 e MQ-6

Sensor	MQ-4	MQ-6
Erro médio (ppm)	826,6	789,4
Desvio Padrão (ppm)	116,2	160,2
Coeficiente de Variação	0,14	0,20

Fonte: Autoria própria

É possível notar que o erro do sistema diminui, em porcentagem, à medida que aumenta a concentração de GLP no ambiente ao qual o sensor foi submetido.

Parte do erro encontrado se deve ao erro humano ao se medir o tempo em que a válvula reguladora ficou aberta e o tempo de abertura e fechamento da válvula. Este erro poderia ser minimizado automatizando o processo por meio da utilização de uma válvula solenoide controlada por um Arduino®

As considerações para o cálculo da densidade média do GLP, também contribuem para o erro na concentração estimada. Além disso foi desconsiderado aumento de pressão no vasilhame ao ser inserido o GLP, sendo necessário um estudo para avaliar a influência deste aumento de pressão no erro.

Como o ambiente dentro do vasilhame não é homogêneo, a leitura da concentração pode variar de acordo com a posição do sensor no interior do vasilhame e isto pode ter contribuído para o erro obtido uma vez que o sensor foi posicionado somente na parte central. Assim, também é necessário um estudo para avaliar a variação das leituras em função da posição do sensor.

Outra fonte possível de erro são os gráficos fornecidos pelo fabricante que, inicialmente, foram desenvolvidos para serem utilizados com uma forma de calibração diferente da que foi proposta neste trabalho.

Também devemos observar que os valores medidos pelos sensores sofrem influência de outros gases além do medido, como já discutido anteriormente, e mesmo ventilando o vasilhame antes das medidas não foi possível garantir que não houvesse a presença de outros gases se não o medido.

5 CONCLUSÃO

- a) Neste trabalho foi desenvolvido uma unidade de monitoramento e controle ambiental, capaz de monitorar a concentração de vários gases de acordo com a necessidade da situação. Utilizando da plataforma Arduino®, dos sensores da família MQ-x e do sensor DHT22, o sistema se mostra mais barato que os sensores e detectores disponíveis no mercado e o custo pode ser reduzido ainda mais utilizando outra plataforma de micro controlador menor e mais barata que o Arduino® MEGA como o ATtiny85 que possui um custo menor, tamanho reduzido e é capaz de utilizar do *software* desenvolvido neste trabalho com poucas alterações.
- b) A metodologia de medição e calibração proposta neste trabalho foi desenvolvida utilizando curvas, fornecidas pelo fabricante dos sensores da família MQ-x, que relacionam a resistência do sensor com a concentração do gás em que o sensor está submetido.
- c) O sistema foi testado utilizando o sensor MQ-6 e o sensor MQ-4 para leitura estimada de cinco níveis de concentração de GLP em um vasilhame de 20,3 litros e o resultado obtido possui um erro médio de 826,6 ppm para o MQ-4 com um desvio padrão de 116,2 ppm e de 789,4 ppm para o MQ-6 com um desvio padrão de 160,2.
- d) Ao comparar o maior erro obtido nesse experimento que foi de 26,9% com o sistema Ventis MX4 da fabricante Industrial Scientific que, segundo seu manual, possui um erro máximo de 5% para a medida da concentração de GLP, podemos concluir que o sistema desenvolvido neste tem uma menor precisão. Entretanto o custo do sistema Ventis MX4 é de R\$ 3950,00 e o do sistema desenvolvido neste trabalho de R\$ 80,00.
- e) O erro encontrado no teste experimental se deve, em parte, às aproximações realizadas no cálculo do volume de GLP inserido e a variação da pressão no interior do vasilhame, que foi desconsiderada nos cálculos. O erro também está relacionado ao erro na medição do tempo em que a válvula reguladora de vazão do GLP esteve aberta e o tempo de abertura e fechamento da válvula. Uma outra fonte possível de erro é a posição do sensor que foi posicionado no centro, uma vez que o ambiente no interior do vasilhame não é homogêneo, diferentes posições podem obter diferentes leituras. Além disso devemos considerar que as curvas que relacionam a resistência do sensor com a concentração do gás em que o sensor está submetido, que foram fornecidas pelo fabricante, foram levantadas para trabalhar com uma forma diferente de calibração da que foi proposta neste trabalho.
- f) Para trabalhos futuros é necessário desenvolver um novo experimento que minimize os erros citados e que permita um maior número de níveis de concentração para que seja possível levantar mais dados e assim realizar um estudo estatístico adequado do sistema. A automatização do experimento por meio de uma válvula solenoide controlada por um segundo Arduino®, seria capaz de minimizar parte do erro obtido no experimento.
- g) Também é necessário um estudo que avalie o efeito nas leituras devido a variação da pressão no interior do vasilhame ao se inserir o GLP.
- h) Além disso, um estudo de como a posição do sensor no interior do vasilhame não homogêneo varia as leituras obtidas pelo sistema, permitiria uma melhor compreensão das características do sistema.

- i) Por fim, o sistema desenvolvido funciona bem como detector para os gases GLP, hidrogênio, dióxido de carbono, monóxido de carbono, álcool, metano e amônia, mas precisa ser mais estudado e aperfeiçoado para ser utilizado como um sistema confiável para a medição da concentração dos gases citados. Porém, apesar das limitações, conclui-se que seja viável a construção de uma unidade de monitoramento de gases em um ambiente que seja de baixo custo e programável.

REFERÊNCIAS

ANP. Agência Nacional do Petróleo, Gás Natural e Biocombustíveis. **Especificações: Gás Natural e GLP**. Brasília, DF, 2017. 6 p.

Corpo de Bombeiros. **Cuidados com o gás de cozinha ou GLP**. Paraná, 2015. Disponível em: <<http://www.bombeiros.pr.gov.br/modules/conteudo/conteudo.php?conteudo=24>>. Acesso em: 22 dez. 2017, 20:51.

F-CHART SOFTWARE. **Engineering Equation Solver**, versão de demonstração 10.269D. Disponível em <<http://www.fchart.com>>. Acesso em: 20 dez. 2017, 19:31

IBP. Instituto Brasileiro de Petróleo, Gás e Biocombustíveis. **Brasil estuda o uso de plantas a gás para geração de energia**. Brasília, 2017. Disponível em: <<https://www.ibp.org.br/noticias/brasil-estuda-o-uso-de-plantas-a-gas-para-geracao-de-energia/>>. Acesso em: 22 dez. 2017, 20:51

LACERDA, A.; LEROUX, T.; MORATA, T. **Efeitos ototóxicos da exposição ao monóxido de carbono: uma revisão**. Pró-Fono Revista de Atualização Científica, Barueri (SP), v. 17, n. 3, p. 403-412, set.-dez. 2005.

MARCICANO, Rodrigo de Lima. **Avaliação e Controle de Ambiente de Trabalho em Depósito de GLP (Gás Liquefeito de Petróleo)**. 2014, 14 p. Centro Universitário de Lins – Unilins. Lins, SP

NAHVI, Mahmood; Edminister, Joseph A. **Circuitos Elétricos**. 5. ed. Porto Alegre, RS. Bookman Editora LTDA. 2014.

RAMOS, Marília Abilio; DROGUETT, Enrique Lopez. **Análise Quantitativa de Risco de um Terminal Offshore de Gás Natural Liquefeito no Porto de Suape**. XXXII ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 2012, Bento Gonçalves, RS.

TILLAART, Rob. **DHTlib**. <<https://github.com/RobTillaart/Arduino/tree/master/libraries/DHTlib>>. Acesso em: 22 dez. 2017, 21:03

VARGAS, Reinaldo A.; CHIBA, Rubens; FRANCO, Egberto G.; SEO, Emília S. M. **Hidrogênio: O Vetor Energético do Futuro?** 2006. 13 p. Instituto de Pesquisas Energéticas e Nucleares – Universidade de São Paulo, São Paulo.

APÊNDICE A - Cálculos dos polinômios do sensor MQ-4

```
%%Início do código para calcular os polinômios do MQ-4
```

```
clear all;close all; clc;
```

```
%%Pontos da curva da resposta ao álcool do sensor MQ-4
```

```
x=[195.8373 218.6686 240.3455 262.0986 283.5781 306.8179 329.3577 359.1671
388.6016 417.1495 444.2813 480.691 520.0846 571.6412 618.4883 674.4663
729.7401 808.4227 861.0034 953.8393 1056.6851 1152.3232 1227.2716 1403.118
1604.1602 1735.6241 1877.8618 2064.0166 2250.826 2435.2855 2634.8618
2918.9602 3183.1487 3416.993 3668.0162 3937.4804 4193.5784 4466.3333
4832.3578 5311.3953 6622.0262 7108.5014 7570.8461 8063.2622 8655.6152
9364.9593 9974.0665];
```

```
y=[8.1492 8 7.9264 7.7812 7.7097 7.5685 7.4988 7.4299 7.2938 7.2938 7.1602
7.1602 7.0291 6.9003 6.774 6.6499 6.4681 6.4086 6.3496 6.176 6.0629 6.0629
6.0071 5.8971 5.7891 5.6831 5.6831 5.579 5.4264 5.327 5.278 5.1337 5.0865 4.9933
4.9474 4.9019 4.8568 4.7678 4.6805 4.6375 4.5106 4.4691 4.428 4.428 4.3873 4.307
4.307];
```

```
%%Passamos os pontos retirados do gráfico em escala logarítmica para
%%escala cartesiana normal
```

```
x1=log10(x);
y1=log10(y);
```

```
%%Comando para encontrar um polinômio capaz de representar a curva no
%%intervalo requerido
```

```
P=polyfit(x1,y1,1);
```

```
%%Avaliamos os mesmo pontos em x para o polinômio para checar como a
%%curva descrita pelo polinômio se aproxima da curva gerada pelos
%%pontos
```

```
yp=polyval(P,x1);
```

```
%%Plotagem da comparação entre a curva gerada pelos pontos e da curva
%%gerada pelo polinômio
```

```
plot(x1,y1,'o');
hold on;
plot(x1,yp,'-');
```

```
%%Pontos da curva da resposta ao CH4 do sensor MQ-4
```

```

x=    [220.2466    240.1331    261.8438    285.5095    311.3227    339.4697
      370.1514    403.6061    440.1086    479.8861    523.2588    570.5672
      622.1527    678.3836    739.6968    806.5955    879.4966    958.9866
      1046.2125    1140.3972    1243.1172    1355.4347    1477.9809    1611.6066
      1757.5052    1916.7167    2090.351    2279.5904    2485.9616    2710.6458
      2955.9599    3223.4748    3515.3918    3833.9537    4180.8133    4558.4317
      4968.395    5418.3314    5906.7577    6442.3749    7020.2389    7653.9015
      8346.8089    9100.9555    9920.2633    10409.3676];
y=    [1.7793    1.7217 1.669 1.6172 1.5677 1.5198 1.4726 1.4269 1.3839 1.3409
      1.2993 1.2595 1.221 1.1831 1.1464 1.1118 1.0773 1.0439 1.0206 0.9834 0.9483
      0.9185 0.8904 0.8631 0.8383 0.8149 0.7922 0.7694 0.7472 0.724 0.7029 0.6823 0.663
      0.6448 0.6256 0.6056 0.5828 0.5663 0.5467 0.5321 0.5102 0.4934 0.4792 0.4641
      0.4472 0.4466];

```

```

x1=log10(x);
y1=log10(y);

```

```

P=polyfit(x1,y1,1);
yp=polyval(P,x1);

```

```

a=P(1)
b=P(2)

```

```

xv=10.^((log10(y)-b)./a);

```

```

figure(1);
plot(x,y,'o');
hold on;
plot(xv,y,'-');
figure(2);
plot(x1,y1,'o');
hold on;
plot(x1,yp,'-');

```

%%Pontos da curva da resposta ao CO do sensor MQ-4

```

x=    [233.9836    255.4133    278.9388    304.5814    332.5812    363.2046
      396.5828    433.0403    472.8492    516.3741    563.9669    615.7614
      672.331    734.1577    801.6918    875.4382    955.8382    1043.8212
      1140.0378    1244.3423    1359.1535    1484.1801    1621.3266    1771.1464
      1934.8102    2113.4823    2307.7096    2520.7486    2753.68    3008.1355
      3286.0145    3589.7586    3921.4725    4282.0862    4675.8615    5105.9871
      5577.8344    6093.0915    6655.0079    7266.994    7938.2886    8669.466
      9465.4082    10336.963    11296.254];
y=    [4.3413    4.2863 4.2665 4.235 4.2037 4.1823 4.1494 4.1188 4.0883 4.0656
      4.0506 4.015 3.9817 3.9541 3.9285 3.9031 3.8688 3.8474 3.8337 3.7796 3.7714 3.747
      3.747 3.747 3.747 3.7435 3.7141 3.709 3.709 3.709 3.7072 3.709 3.709
      3.6833 3.6577 3.6341 3.6343 3.6326 3.6223 3.5972 3.5955 3.5789 3.5458 3.5278
      3.5496];

```

```
x1=log10(x);
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
yp=polyval(P,x1);
```

```
a=P(1)
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
plot(x,y,'o');
hold on;
plot(xv,y,'-');
figure(2);
plot(x1,y1,'o');
hold on;
plot(x1,yp,'-');
```

```
%%Pontos da curva da resposta ao H2 do sensor MQ-4
```

```
x=[1138.2393 5526.5465 6605.5484 7718.9816 8900.3570 9952.6532 10874.8259 4453.0637
3353.1442 2243.5424 1010.8302 903.9549 790.7797 681.4410 573.7901 452.5248 346.1112
231.4481];
y=[2.8063 2.1682 2.0869 2.0087 1.9896 1.9519 1.8967 2.2527 2.3404 2.5263 2.8879 2.9436
3.0292 3.1173 3.1774 3.3649 3.5295 3.7737];
```

```
x1=log10(x);
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
yp=polyval(P,x1);
```

```
a=P(1)
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
plot(x,y,'o');
hold on;
plot(xv,y,'-');
figure(2);
plot(x1,y1,'o');
hold on;
plot(x1,yp,'-');
```

```
%%Pontos da curva da resposta ao GLP do sensor MQ-4
```

```

x= [245.6117 267.8396 292.0712 318.495 347.3189 378.7307
413.0059 450.3584 491.1293 535.5912 584.0942 636.9722
694.6563 757.5848 826.1464 900.9619 982.5796 1071.3571
1168.7933 1274.6041 1390.0698 1515.7473 1652.8326 1802.4143
1965.694 2143.8237 2337.8401 2549.6238 2781.1997 3033.6435
3308.5498 3608.4664 3935.3554 4291.3888 4680.1433 5104.3934
5569.0758 6074.7351 6625.2228 7225.9893 7879.9429 8597.7686
9378.9379 10229.4078 10732.8758 226.3031];
y= [2.5264 2.4525 2.3797 2.309 2.2415 2.174 2.1104 2.0468 1.9878 1.9306
1.8759 1.8219 1.7703 1.7209 1.6706 1.6232 1.5779 1.5282 1.4939 1.4509 1.4104
1.3673 1.3261 1.2873 1.2514 1.2171 1.1815 1.1485 1.1206 1.0924 1.0624 1.0338
1.0049 0.9751 0.9479 0.9223 0.9029 0.8805 0.8564 0.8336 0.8093 0.7929 0.774
0.7535 0.7514 2.5751];

```

```

x1=log10(x);
y1=log10(y);

```

```

P=polyfit(x1,y1,1);
yp=polyval(P,x1);

```

```

a=P(1)
b=P(2)

```

```

xv=10.^((log10(y)-b)./a);

```

```

figure(1);
plot(x,y,'o');
hold on;
plot(xv,y,'-');
figure(2);
plot(x1,y1,'o');
hold on;
plot(x1,yp,'-');

```

%%Pontos da curva da resposta a umidade e temperatura do sensor MQ-4

```

x= [-12.075 -10.642 -9.209 -7.738 -6.347 -5.007 -4.912 -3.375 -2.051 -
0.623 0.801 2.274 3.643 5.143 6.61 7.982 9.427 10.788 12.273 13.714 15.188
16.452 17.947 19.338 20.74 22.178 23.35 24.924 26.475 27.9 29.112 30.885
32.063 33.57 34.995 36.449 37.874 39.283 40.754 42.682 43.753 45.464 47.015
48.515 49.588];
y= [1.212 1.197 1.183 1.169 1.155 1.135 1.139 1.124 1.112 1.099 1.087 1.075
1.065 1.057 1.048 1.04 1.034 1.029 1.025 1.022 1.015 1.01 1.006 1.005
1 0.996 0.995 0.993 0.99 0.986 0.984 0.981 0.981 0.978 0.975 0.972
0.968 0.965 0.961 0.958 0.959 0.956 0.953 0.948 0.947];

x2= [-11.006 -9.586 -8.166 -6.749 -5.327 -3.9 -2.485 -1.064 0.357 1.776 3.197
4.618 6.022 7.423 8.825 10.227 11.627 13.029 14.43 15.833 17.233 18.637
20.034 21.429 22.823 24.217 25.613 27.007 28.401 29.795 31.19 32.584 33.978
35.373 36.769 38.163 39.558 40.952 42.346 43.741 45.135 46.53 47.925 49.194];

```

```

y2= [1.073 1.062 1.052 1.042 1.031 1.019 1.01 0.999 0.988 0.978 0.967 0.956
      0.95 0.946 0.941 0.936 0.932 0.927 0.923 0.918 0.913 0.908 0.905 0.902
      0.9 0.898 0.895 0.892 0.89 0.887 0.885 0.883 0.88 0.878 0.875 0.872
      0.87 0.868 0.865 0.863 0.86 0.858 0.855 0.853];

```

```

P=polyfit(x,y,2);
yp=polyval(P,x);
P2=polyfit(x2,y2,2);
yp2=polyval(P2,x2);

```

```

P1=[0.00008 -0.0067 1.1024];
P2=[0.00006 -0.0057 0.9941];

```

```

xli=-10:1:50;
yli=polyval(P,xli);
yli2=polyval(P2,xli);

```

```

%%Plot
figure(1);
%plot(x,y,'o');
hold on;
%plot(x,yp,'-');
%plot(x2,y2,'o');
% plot(x2,yp2,'-');
plot(xli,yli,'-');
plot(xli,yli2,'-');
axis([-10 50 0.5 1.4]);
xlabel('°C');
ylabel('Rs/Ro');
legend('33%RH','85%RH');
grid on;

```

```

%%Final do código

```

APÊNDICE B - Cálculos dos polinômios do sensor MQ-6

```
%%Início do código para calcular os polinômios do MQ-6
```

```
clear all;close all; clc;
```

```
%%Pontos da curva da resposta ao álcool do sensor MQ-6
```

```
x= [195.8373    218.6686    240.3455    262.0986    283.5781    306.8179
    329.3577    359.1671    388.6016    417.1495    444.2813    480.691
    520.0846    571.6412    618.4883    674.4663    729.7401    808.4227
    861.0034    953.8393    1056.6851    1152.3232    1227.2716    1403.118
    1604.1602    1735.6241    1877.8618    2064.0166    2250.826    2435.2855
    2634.8618    2918.9602    3183.1487    3416.993    3668.0162    3937.4804
    4193.5784    4466.3333    4832.3578    5311.3953    6622.0262    7108.5014
    7570.8461    8063.2622    8655.6152    9364.9593    9974.0665];
y= [8.1492      8      7.9264 7.7812 7.7097 7.5685 7.4988 7.4299 7.2938 7.2938
    7.1602 7.1602 7.0291 6.9003 6.774  6.6499 6.4681 6.4086 6.3496 6.176  6.0629
    6.0629 6.0071 5.8971 5.7891 5.6831 5.6831 5.579  5.4264 5.327  5.278  5.1337
    5.0865 4.9933 4.9474 4.9019 4.8568 4.7678 4.6805 4.6375 4.5106 4.4691 4.428  4.428
    4.3873 4.307  4.307];
```

```
x1=log10(x);
```

```
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
```

```
yp=polyval(P,x1);
```

```
a=P(1)
```

```
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
```

```
loglog(x,y,'o');
```

```
hold on;
```

```
grid on;
```

```
loglog(xv,y,'-');
```

```
axis([100 10000 0.1 10]);
```

```
title('MQ-6');
```

```
xlabel('ppm');
```

```
Flabele('Räs/Roo');
```

```
%%Pontos da curva da resposta ao CH4 do sensor MQ-6
```

```
X= [195.8373    215.2509    234.7327    255.9779    281.3533    309.2441
    342.5877    373.5946    404.2114    440.7956    496.0771    549.5656
    604.0447    663.9244    724.0147    795.7872    881.5913    1007.9078
    1143.2824    1276.5701    1403.118    1579.0873    1777.1254    1968.7402
    2181.0155    2416.1789    2634.8618    2873.3371    3060.2221    3337.1959
    3610.6854    3906.5879    4226.7402    4609.2929    5066.2178    5612.4721
    6120.4442    6674.3917    7336.0323    8127.0247    8932.6666    9513.6569
    10293.3192];
```

```
Y= [2.5907      2.5198 2.4509 2.362  2.2763 2.1735 2.0946 2.0186 1.9453 1.8921
    1.8067 1.7411 1.6779 1.617  1.5583 1.5018 1.4208 1.3441 1.2716 1.2368 1.192
    1.1381 1.0767 1.0377 0.9908 0.9727 0.9374 0.9117 0.895  0.8546 0.8236 0.7937
    0.7649 0.7304 0.717  0.6783 0.6598 0.6358 0.6127 0.5905 0.5743 0.5586 0.5384];
```

```
X1=log10(x);
Y1=log10(y);
```

```
P=polyfit(x1,y1,1);
yp=polyval(P,x1);
```

```
a=P(1)
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
plot(x,y,'o');
hold on;
plot(xv,y,'-');
figure(2);
plot(x1,y1,'o');
hold on;
plot(x1,yp,'-');
```

%%Pontos da curva da resposta ao CO do sensor MQ-6

```
x= [195.8373    208.5747    227.4524    246.0925    268.3658    290.3589
    316.6386    345.2969    376.5489    417.1495    451.3357    496.0771
    536.7316    589.9385    633.2772    679.7998    735.5107    795.7872
    847.546     902.6714    961.3821   1032.0083   1099.1312   1189.2071
   1266.5544   1359.5996   1436.6686   1530.1111   1629.6311   1749.3491
   1892.7115   2015.8156   2163.9038   2304.6465   2473.9532   2614.1893
   2828.4271   3084.4217   3311.0131   3582.3569   3875.9379   4193.5784
   4573.1296   4909.0864   5269.7235   5656.8542   6072.4248   6674.3917
   7221.3708   7751.8758   8191.2913   8655.6152   9364.9593
  10132.4355];
y= [9.0213      9.0213 8.8561 8.9383 8.8561 8.7746 8.7746 8.6139 8.6139 8.6139
    8.4561 8.4561 8.4561 8.3784 8.3013 8.2249 8.2249 8.1492 8      7.9264 7.9264
    7.9264 7.9264 7.7812 7.7097 7.7097 7.7097 7.7812 7.7812 7.7812 7.6387 7.6387
    7.6387 7.5685 7.4299 7.4988 7.4299 7.3615 7.3615 7.2938 7.2938 7.2267 7.1602
    7.0943 7.0943 6.9644 6.9644 6.9003 6.8369 6.7116 6.7116 6.6499 6.6499 6.5887];
```

```
x1=log10(x);
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
yp=polyval(P,x1);
```

```
a=P(1)
```

```
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
plot(x,y,'o');
hold on;
plot(xv,y,'-');
figure(2);
plot(x1,y1,'o');
hold on;
plot(x1,yp,'-');
```

```
%%Pontos da curva da resposta ao H2 do sensor MQ-6
```

```
x=    [194.3008    213.5621    231.0639    246.0925    266.2603    285.8206
      304.4107    324.2099    342.5877    364.8738    388.6016    413.8766
      437.3373    469.4655    492.185    524.1973    558.2917    599.3055
      638.285    696.0548    759.0531    821.2589    888.5627    984.3701
      1081.9519    1143.2824    1227.2716    1317.4309    1482.6539    1629.6311
      1735.6241    1907.6787    2064.0166    2268.625    2454.5432    2614.1893
      2784.2191    3060.2221    3259.2623    3668.0162    4000    4431.2916
      4832.3578    5228.3787    6780.3683    7394.044    8000    8724.0619
      9664.7156];
y=    [5.7891    5.6308    5.5277    5.3765    5.2295    5.0865    4.9933    4.9019    4.8121    4.6805
      4.5948    4.5106    4.428    4.3469    4.2673    4.1892    4.0746    4.0746    3.9267    3.8194    3.7149
      3.5801    3.5145    3.387    3.325    3.2641    3.2043    3.1167    3.0314    2.9485    2.8945    2.8154
      2.7638    2.6635    2.6147    2.5669    2.4967    2.4509    2.4284    2.3839    2.3187    2.3187    2.2974
      2.2553    2.1735    2.1535    2.114    2.114    2.0753];
```

```
x1=log10(x);
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
yp=polyval(P,x1);
```

```
a=P(1)
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
plot(x,y,'o');
hold on;
plot(xv,y,'-');
figure(2);
plot(x1,y1,'o');
hold on;
plot(x1,yp,'-');
```

%%Pontos da curva da resposta ao GLP do sensor MQ-6

%%Valores do gráfico Datasheet

```
x= [192.084    210.111    230.382    253.824    278.107    303.31
    332.052    366.143    402.283    440.769    484.544    529.138
    581.689    634.17    694.84    762.58    834.148    923.105    1008.062
    1104.731    1217.982    1330.354    1461.668    1597.076    1749.865
    1917.272    2095.466    2313.163    2548.179    2812    3078.177
    3375.775    3695.318    4052.581    4439.874    4876.766    5325.595
    5849.644    6403.948    7063.381    7746.269    8500.667    9290.732
    10179.561    10744.242];
y= [2.051 1.983 1.906 1.828 1.759 1.694 1.628 1.556 1.491 1.429 1.364 1.32
    1.251 1.209 1.165 1.111 1.074 1.032 0.998 0.958 0.921 0.891 0.854 0.825
    0.791 0.75 0.729 0.698 0.672 0.65 0.631 0.604 0.58 0.56 0.542 0.519
    0.503 0.485 0.468 0.448 0.433 0.42 0.406 0.391 0.387];
```

```
x1=log10(x);
```

```
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
```

```
yp=polyval(P,x1);
```

```
yv=10.^((-0.4189*log10(x))+1.2616);
```

```
xv=10.^((1.2616-log10(y))/0.4189);
```

```
figure(1)
```

```
plot(x,y,'o');
```

```
hold on;
```

```
plot(x,yv,'-');
```

```
grid on;
```

```
plot(xv,y,'-r');
```

```
figure(2);
```

```
plot(x1,y1,'s');
```

```
hold on;
```

```
plot(x1,yp,'-');
```

```
grid on;
```

%%Pontos da curva da resposta a umidade e temperatura do sensor MQ-6

```
x= [-12.075    -10.642    -9.209 -7.738 -6.347 -5.007 -4.912 -3.375 -2.051 -
    0.623 0.801 2.274 3.643 5.143 6.61 7.982 9.427 10.788 12.273 13.714 15.188
    16.452 17.947 19.338 20.74 22.178 23.35 24.924 26.475 27.9 29.112 30.885
    32.063 33.57 34.995 36.449 37.874 39.283 40.754 42.682 43.753 45.464 47.015
    48.515 49.588];
```

```
y= [1.212 1.197 1.183 1.169 1.155 1.135 1.139 1.124 1.112 1.099 1.087 1.075
    1.065 1.057 1.048 1.04 1.034 1.029 1.025 1.022 1.015 1.01 1.006 1.005
```

```

1      0.996 0.995 0.993 0.99  0.986 0.984 0.981 0.981 0.978 0.975 0.972
0.968 0.965 0.961 0.958 0.959 0.956 0.953 0.948 0.947];

```

```

x2=   [-11.006      -9.586 -8.166 -6.749 -5.327 -3.9   -2.485 -1.064 0.357  1.776  3.197
      4.618  6.022  7.423  8.825 10.227 11.627 13.029 14.43  15.833 17.233 18.637
      20.034 21.429 22.823 24.217 25.613 27.007 28.401 29.795 31.19  32.584 33.978
      35.373 36.769 38.163 39.558 40.952 42.346 43.741 45.135 46.53  47.925 49.194];

```

```

y2=   [1.073 1.062 1.052 1.042 1.031 1.019 1.01  0.999 0.988 0.978 0.967 0.956
      0.95  0.946 0.941 0.936 0.932 0.927 0.923 0.918 0.913 0.908 0.905 0.902
      0.9   0.898 0.895 0.892 0.89  0.887 0.885 0.883 0.88  0.878 0.875 0.872
      0.87  0.868 0.865 0.863 0.86  0.858 0.855 0.853];

```

```

P=polyfit(x,y,2);

```

```

yp=polyval(P,x);

```

```

P2=polyfit(x2,y2,2);

```

```

yp2=polyval(P2,x2);

```

```

P1=[0.00008 -0.0067 1.1024];

```

```

P2=[0.00006 -0.0057 0.9941];

```

```

xli=-10:1:50;

```

```

yli=polyval(P,xli);

```

```

yli2=polyval(P2,xli);

```

```

%%Plot

```

```

figure(1);

```

```

%plot(x,y,'o');

```

```

hold on;

```

```

%plot(x,yp,'-');

```

```

%plot(x2,y2,'o');

```

```

% plot(x2,yp2,'-');

```

```
plot(xli,yli,'-');  
plot(xli,yli2,'-');  
axis([-10 50 0.5 1.4]);  
xlabel('°C');  
ylabel('Rs/Ro');  
legend('33%RH','85%RH');  
grid on;
```

ANEXO C - Cálculos dos polinômios do sensor MQ-8

%%Início do código para calcular os polinômios do MQ-8

clear all;close all; clc;

%%Pontos da curva da resposta ao álcool do sensor MQ-8

```
x= [204.1821    230.0074    261.1185    298.6257    344.393    390.8961
    436.7539    507.6161    576.0413    664.3253    760.0599    876.5466
    1003.2739    1148.0884    1313.2689    1454.7985    1677.7607    1919.5397
    2161.8853    2473.4306    2806.2686    3109.9675    3501.8915    4038.591
    4844.8484    6914.2212    7600.1304    8554.4175    9480.1902];
y= [24.0797    23.2388    22.0322    19.8036    18.4446    17.1789
    16.2869    14.902    13.6349    12.6992    11.8277    11.0161
    10.6314    10.0794    9.2223    8.2895    7.7206    7.1908    6.8174    6.3496    5.7073
    5.3157    4.9509    4.6112    4.1447    3.2317    2.9569    2.6578    2.4755];
```

x1=log10(x);

y1=log10(y);

P=polyfit(x1,y1,1);

yp=polyval(P,x1);

a=P(1)

b=P(2)

xv=10.^((log10(y)-b)./a);

figure(1);

loglog(x,y,'o');

hold on;

loglog(xv,y,'-');

axis([100 10000 0.01 100]);

grid on;

clear all;

```
x= [201.7358    225.1723    247.4089    274.0718    298.7477    325.7119
    357.804    393.2188    435.4176    501.637    555.4708    629.8318
    708.4804    803.1608    888.9898    1000    1116.1738    1216.6679
    1336.8183    1503.1357    1664.4468    1828.817    1993.4735    2189.8889
    2405.6569    2663.8233    2973.8973    3343.2053    3730.8361    4228.5559
    4944.8404    5515.92    6202.1712    7026.7131    8026.1916    8813.4043
    9757.2335];
y= [8.4381    7.3198    6.4635    5.8097    5.1301    4.6112    4    3.5954    3.1189    2.6578
    2.3056    1.9648    1.6743    1.4017    1.1735    1    0.8675    0.766    0.6764    0.5563    0.4825
    0.4261    0.3762    0.3264    0.2831    0.2456    0.2169    0.1752    0.1493    0.1228    0.0957    0.0787
    0.0648    0.0514    0.0415    0.0348    0.0296];
```

x1=log10(x);

y1=log10(y);

```
P=polyfit(x1,y1,1);
yp=polyval(P,x1);
```

```
a=P(1)
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
loglog(x,y,'o');
hold on;
loglog(xv,y,'-');
```

```
%%Pontos da curva da resposta ao CH4 do sensor MQ-8
```

```
x=[281.1143 690.0214 1095.4872 1358.4384 2013.6991 2674.3444 3958.1744 6437.7472
12673.9301];
y=[25.8216 15.2968 11.1597 10.3278 7.8228 6.9730 5.0981 3.7830 2.1395];
```

```
x1=log10(x);
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
yp=polyval(P,x1);
```

```
a=P(1)
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
plot(x,y,'o');
hold on;
plot(xv,y,'-');
figure(2);
plot(x1,y1,'o');
hold on;
plot(x1,yp,'-');
```

```
%%Pontos da curva da resposta ao CO do sensor MQ-8
```

```
x=[291.3308 725.6305 1162.3298 1445.4191 2157.1978 2873.5564 4286.3597 7128.6973
14133.6461];
y=[57.0929 45.7575 45.2333 43.0992 40.1617 38.8270 35.7536 32.0040 27.1051];
```

```
x1=log10(x);
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
```

```
yp=polyval(P,x1);
```

```
a=P(1)
```

```
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
```

```
plot(x,y,'o');
```

```
hold on;
```

```
plot(xv,y,'-');
```

```
figure(2);
```

```
plot(x1,y1,'o');
```

```
hold on;
```

```
plot(x1,yp,'-');
```

```
%%Pontos da curva da resposta ao H2 do sensor MQ-8
```

```
x= [201.7358    225.1723    247.4089    274.0718    298.7477    325.7119
    357.804    393.2188    435.4176    501.637    555.4708    629.8318
    708.4804    803.1608    888.9898    1000    1116.1738    1216.6679
    1336.8183    1503.1357    1664.4468    1828.817    1993.4735    2189.8889
    2405.6569    2663.8233    2973.8973    3343.2053    3730.8361    4228.5559
    4944.8404    5515.92    6202.1712    7026.7131    8026.1916    8813.4043
    9757.2335];
y= [8.4381    7.3198 6.4635 5.8097 5.1301 4.6112 4    3.5954 3.1189 2.6578
    2.3056 1.9648 1.6743 1.4017 1.1735 1    0.8675 0.766    0.6764 0.5563 0.4825
    0.4261 0.3762 0.3264 0.2831 0.2456 0.2169 0.1752 0.1493 0.1228 0.0957 0.0787
    0.0648 0.0514 0.0415 0.0348 0.0296];
```

```
x1=log10(x);
```

```
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
```

```
yp=polyval(P,x1);
```

```
a=P(1)
```

```
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
```

```
plot(x,y,'o');
```

```
hold on;
```

```
plot(xv,y,'-');
```

```
figure(2);
```

```
plot(x1,y1,'o');
```

```
hold on;
```

```
plot(x1,yp,'-');
```

%%Pontos da curva da resposta ao GLP do sensor MQ-8

```
x= [203.8435    228.9047    257.0628    288.6527    324.1126    363.9736
     408.7418    458.9542    515.4305    578.7992    650.1112    730.3715
     820.419     921.4546   1034.8818   1162.3432   1305.5195   1466.2415
    1646.9534   1850.2118   2077.8881   2334.3591   2622.1621   2945.0483
    3308.3474   3716.5546   4175.3355   4691.0391   5269.5923   5919.2801
    6650.2173   7471.6904   8393.6002   9428.9125  10480.4371];
y= [35.1852    33.4529    31.9771    30.2724    28.5663    27.2476
     26.0176    24.5513    23.544    22.3848    21.7216    21.4895
     20.9876    20.2784    19.5092    18.8702    18.2718    17.5975
     17.1313    16.8937    16.2006    15.9931    15.6195    15.0755
     14.8027    14.5661    14.395    14.3025    14.0136    13.6863
     13.5692    13.4965    13.2808    13.0264    12.7907];
```

```
x1=log10(x);
```

```
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
```

```
yp=polyval(P,x1);
```

```
a=P(1)
```

```
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
```

```
plot(x,y,'o');
```

```
hold on;
```

```
plot(xv,y,'-');
```

```
figure(2);
```

```
plot(x1,y1,'o');
```

```
hold on;
```

```
plot(x1,yp,'-');
```

%%Pontos da curva da resposta a umidade e temperatura do sensor MQ-8

```
x=[-12.7498 -7.5327 -2.3066 2.4257 8.1087 13.3953 18.2847 23.7411 28.9483 34.0706
    39.0738 44.2909 49.6651];
```

```
y=[1.7005 1.6136 1.5798 1.4901 1.4149 1.2957 1.2472 1.1633 1.1382 1.0955 0.9941 0.9072
    0.8615];
```

```
x2=[];
```

```
y2=[];
```

```
P1=polyfit(x,y,2);
```

```
yp=polyval(P1,x);
```

```
P2=polyfit(x2,y2,2);
```

```
yp2=polyval(P2,x2);
```

```
P1=[0.00012 -0.00656 0.98828];  
P2=[0.00014 -0.00753 1.04125];
```

```
xli=-10:1:50;  
yli=polyval(P1,xli);  
yli2=polyval(P2,xli);
```

```
%%Plot  
figure(1);  
plot(x,y,'o');  
hold on;  
%plot(x,yp,'-');  
plot(x2,y2,'o');  
% plot(x2,yp2,'-');  
plot(xli,yli,'-r');  
plot(xli,yli2,'-b');  
axis([-10 50 0.1 1.3]);  
grid on;
```

APÊNDICE D - Cálculos dos polinômios do sensor MQ-135

```
%%Início do código para calcular os polinômios do MQ-135
```

```
clear all;close all; clc;
```

```
%%Pontos da curva da resposta ao CO do sensor MQ-135
```

```
x=    [198.4507    100.1526    88.9166    79.5364    69.057 59.0758
      49.4207    39.5402    29.5883    19.8044    9.9188];
y=    [1.3434    1.5677 1.6438 1.6792 1.7622 1.8345 1.9444 2.0822 2.2546 2.5372
      3.1452];
```

```
x1=log10(x);
```

```
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
```

```
yp=polyval(P,x1);
```

```
a=P(1)
```

```
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
```

```
plot(x,y,'o');
```

```
hold on;
```

```
plot(xv,y,'-');
```

```
figure(2);
```

```
plot(x1,y1,'o');
```

```
hold on;
```

```
plot(x1,yp,'-');
```

```
%%Pontos da curva da resposta ao CO2 do sensor MQ-135
```

```
x=    [198.781    147.6286    114.6398    100.2748    89.6914    80.2228
      68.6239    59.5807    49.843 39.2878    29.8362    19.8203
      9.852];
y=    [0.8022    0.907 1.0151 1.0741 1.1163 1.1702 1.2393 1.3118 1.3904 1.5156
      1.6975 1.9776 2.5598];
```

```
x1=log10(x);
```

```
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
```

```
yp=polyval(P,x1);
```

```
a=P(1)
```

```
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
loglog(x,y,'o');
hold on;
loglog(xv,y,'-');
clear all;
```

```
x=    [198.4507    100.1526    88.9166    79.5364    69.057 59.0758
      49.4207    39.5402    29.5883    19.8044    9.9188];
y=    [1.3434    1.5677 1.6438 1.6792 1.7622 1.8345 1.9444 2.0822 2.2546 2.5372
      3.1452];
```

```
x1=log10(x);
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
yp=polyval(P,x1);
```

```
a=P(1)
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

%%Pontos da curva da resposta ao NH4 do sensor MQ-135

```
x=    [197.3426    130.1144    99.5492    89.0374    79.6379    68.6296
      59.143 49.4741    39.2856    29.8338    19.8176    17.2056
      9.7758];
y=    [0.7621    0.9113 1.0204 1.0789 1.131  1.2078 1.2898 1.3908 1.5418 1.7418
      2.0644 2.204  2.8386];
```

```
x1=log10(x);
y1=log10(y);
```

```
P=polyfit(x1,y1,1);
yp=polyval(P,x1);
```

```
a=P(1)
b=P(2)
```

```
xv=10.^((log10(y)-b)./a);
```

```
figure(1);
loglog(x,y,'o');
hold on;
loglog(xv,y,'-');
clear all;
```

%%Pontos da curva da resposta a umidade e temperatura do sensor MQ-135

```
x= [-10.0767 -8.2027 -6.4139 -4.2843 -2.4103 -0.5363
    1.3377 3.2117 5.0857 6.9597 8.8337 10.7077 12.5817 14.4557
    16.3298 18.2038 20.0778 21.9518 23.8258 25.6998
    27.7442 29.6182 31.4922 33.5366 35.4106 37.3414
    39.1586 41.0326 42.9407 44.7806 46.825 48.699 50.2323];
```

```
y= [1.6967 1.6404 1.5852 1.5142 1.4538 1.3956 1.3408 1.2928 1.2478 1.2032
    1.1611 1.1235 1.0895 1.0598 1.0368 1.0218 1.008 0.9956 0.9857 0.9814 0.9742
    0.9706 0.9677 0.9605 0.9613 0.9567 0.9548 0.9556 0.9507 0.9491 0.9419 0.9386
    0.9352];
```

```
x2= [-10.0767 -8.2027 -6.4801 -4.2843 -2.4103 -0.5363
    1.3377 3.2117 5.0857 6.9597 8.8337 10.7077 12.5817 14.4557
    16.3298 18.2038 20.0778 21.9518 23.8258 25.6998
    27.5738 29.4478 31.3218 33.1958 35.0698 36.9439
    38.8179 40.6919 42.5659 44.4399 46.3139 48.1879
    49.9767];
```

```
y2= [1.533 1.477 1.4286 1.3684 1.3161 1.2642 1.2144 1.1723 1.1343 1.0897 1.0575
    1.0264 0.9939 0.9697 0.9519 0.9358 0.9216 0.911 0.9008 0.8961 0.8924 0.8866
    0.8827 0.8794 0.8736 0.8703 0.866 0.8609 0.8584 0.853 0.8501 0.8457 0.8388];
```

```
P1=polyfit(x,y,3)
```

```
yp=polyval(P1,x);
```

```
P2=polyfit(x2,y2,3)
```

```
yp2=polyval(P2,x2);
```

```
P1=[-0.000005 0.000653 -0.028508 1.366829];
```

```
P2=[-0.0000046 0.000573 -0.025026 1.2384997];
```

```
xli=-10:1:50;
```

```
yli=polyval(P1,xli);
```

```
yli2=polyval(P2,xli);
```

```
%%Plot
```

```
figure(1);  
%plot(x,y,'o');  
hold on;  
%plot(x,yp,'-');  
%plot(x2,y2,'o');  
% plot(x2,yp2,'-');  
plot(xli,yli,'-r');
```

APÊNDICE E - Código para Arduino

```

#include <LiquidCrystal.h>

#include "DHT.h"

#include <math.h>

#define sub_menu_index_max_value 6      // Valor máximo que o index dos sub-menus
pode assumir (0,1,2,...,sub_menu_index_max_value)

#define menu_index_max_value 1          // Valor máximo que o index dos menus pode
assumir (0,1,2,...,menu_index_max_value)

#define SENSOR_ANALOG_PORT 12          // Pino da porta analogica para o sensor de
gas

#define DHTPIN 35                      // Pino da porta digital para o sensor de temperatura e
umidade

#define DHTTYPE DHT22                  // Tipo do sensor de temperatura e umidade

#define VOLT_STEP 0.004883             // Voltagem na leitura da porta analogica 5/1023

#define RL_MQ6 10000                   // Valor da resistência de carga para o sensor MQ6
#define RL_MQ8 10000                   // Valor da resistência de carga para o sensor MQ8
#define RL_MQ4 10000                   // Valor da resistência de carga para o sensor MQ4
#define RL_MQ135 10000                 // Valor da resistência de carga para o sensor MQ135

#define R_STANDARD_MQ6 31894           // Valor de R0 para o sensor MQ6 padrão
#define R_STANDARD_MQ8 17900           // Valor de R0 para o sensor MQ8 padrão
#define R_STANDARD_MQ4 22375           // Valor de R0 para o sensor MQ4 padrão
#define R_STANDARD_MQ135 27338         // Valor de R0 para o sensor MQ135 padrão

#define CALIBRATION_SAMPLING_TIME 100  // Valor do tempo de amostragem para
a calibração em [ms]

#define NUMBER_OF_CALIBRATION_SAMPLE 1000 // Quantidade de amostras para a
calibração

#define HEAT_TIME 86400                // Tempo de aquecimento em segundos

```

```
#define right 1           // Valor relativo ao botão direito do display LCD
#define up 2             // Valor relativo ao botão superior do display LCD
#define down 3           // Valor relativo ao botão inferior do display LCD
#define left 4           // Valor relativo ao botão esquerdo do display LCD
#define select 5         // Valor relativo ao botão de seleção do display LCD
```

```
/******Sensores presentes******/
```

```
#define MQ6 0
```

```
#define MQ8 1
```

```
#define MQ4 2
```

```
#define MQ135 3
```

```
/******Gas presentes******/
```

```
#define ALCOOL 0
```

```
#define H2 1
```

```
#define LPG 2
```

```
#define CO 3
```

```
#define CO2 4
```

```
#define CH4 5
```

```
#define SMOKE 6
```

```
#define NH4 7
```

```
int i_selected_config = 0;
```

```
int i_sensor_analog_reading = 0;
```

```
float R0 = 0;
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
```

```
void setup()
{
  lcd.begin(16, 2);
  Serial.begin(9600);
  Configuration_Menu();
  dht.begin();
}

void loop()
{
  float temperature, humidity;
  float RS = 0;
  float rs_r0_ratio = 0;
  float PPM = 0;
  humidity = dht.readHumidity();
  temperature = dht.readTemperature();
  delay(2000);
  //lcd.clear();
  char tempF[6];
  char humF[6];
  dtostrf(temperature, 5, 1, tempF);
  dtostrf(humidity, 2, 0, humF);
  lcd.setCursor(0, 1);
  lcd.print("T:");
  lcd.print(tempF);
  lcd.print((char)223);
  lcd.print("C ");
  lcd.print("H: ");
  lcd.print(humF);
  lcd.print("%");
```

```

//Faz a leitura de acordo com o configurado inicialmente
switch (i_selected_config)
{
    /*******MQ-6|LPG*****/
    case 0:
        i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
        RS = RL_MQ6 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
        R0 = 19000;
        rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);
        PPM = pow(10, ((-log10(rs_r0_ratio) + 1.2616) / 0.4189));
        lcd.setCursor(0, 0);
        lcd.print(PPM);
        lcd.print(" PPM de LPG");
        break;
    /*******MQ-6|H2*****/
    case 1:
        i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
        RS = RL_MQ6 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
        //R0 = 19000;
        rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);
        PPM = pow(10, ((-log10(rs_r0_ratio) + 2.6292) / 0.6942));
        lcd.setCursor(0, 0);
        lcd.print(PPM);
        lcd.print(" PPM de H2");
        break;
    /*******MQ-6|CO*****/
    case 2:
        i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
        RS = RL_MQ6 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);

```

```

//R0 = 19000;

rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);

PPM = pow(10, ((-log10(rs_r0_ratio) + 2.4024) / 0.5002));

lcd.setCursor(0, 0);

lcd.print(PPM);

lcd.print(" PPM de CO");

break;

/*****MQ-6|CH4*****/

case 3:

i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);

RS = RL_MQ6 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);

//R0 = 19000;

rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);

PPM = pow(10, ((-log10(rs_r0_ratio) + 2.6002) / 0.8233));

lcd.setCursor(0, 0);

lcd.print(PPM);

lcd.print(" PPM de CH4");

break;

/*****MQ-6|ALCOOL *****/

case 4:

i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);

RS = RL_MQ6 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);

//R0 = 19000;

rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);

PPM = pow(10, ((-log10(rs_r0_ratio) + 2.5652) / 0.5905));

lcd.setCursor(0, 0);

lcd.print(PPM);

lcd.print(" PPM de ALCOOL");

break;

/*****MQ-8|LPG*****/

```

case 5:

```
i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
RS = RL_MQ8 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
//R0 = 19000;
rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);
PPM = pow(10, ((-log10(rs_r0_ratio) + 2.21135) / 0.28549));
lcd.setCursor(0, 0);
lcd.print(PPM);
lcd.print(" PPM de LPG");
break;
/*****MQ-8|H2*****/
```

case 6:

```
i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
RS = RL_MQ8 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
//R0 = 19000;
rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);
PPM = pow(10, ((-log10(rs_r0_ratio) + 4.8965) / 1.59295));
lcd.setCursor(0, 0);
lcd.print(PPM);
lcd.print(" PPM de H2");
break;
/*****MQ-8|CO*****/
```

case 7:

```
i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
RS = RL_MQ8 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
//R0 = 19000;
rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);
PPM = pow(10, ((-log10(rs_r0_ratio) + 2.2559) / 0.15833));
lcd.setCursor(0, 0);
lcd.print(PPM);
```

```

    lcd.print(" PPM de CO");

    break;

/*****MQ-8|CH4*****/

case 8:

    i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
    RS = RL_MQ8 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
    //R0 = 19000;

    rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);
    PPM = pow(10, ((-log10(rs_r0_ratio) + 2.20549) / 0.18206));
    lcd.setCursor(0, 0);
    lcd.print(PPM);
    lcd.print(" PPM de CH4");

    break;

/*****MQ-8|ALCOOL *****/

case 9:

    i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
    RS = RL_MQ8 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
    //R0 = 19000;

    rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);
    PPM = pow(10, ((-log10(rs_r0_ratio) + 3.00877) / 0.64202));
    lcd.setCursor(0, 0);
    lcd.print(PPM);
    lcd.print(" PPM de ALCOOL");

    break;

    /*****MQ-4|LPG*****/

case 10:

    i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
    RS = RL_MQ4 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
    //R0 = 19000;

    rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);

```

```

PPM = pow(10, ((-log10(rs_r0_ratio) + 1.17318) / 0.32514));
lcd.setCursor(0, 0);
lcd.print(PPM);
lcd.print(" PPM de LPG");
break;

/*****MQ-4|H2*****/
case 11:
    i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
    RS = RL_MQ4 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
    //R0 = 19000;
    rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);
    PPM = pow(10, ((-log10(rs_r0_ratio) + 0.99150) / 0.17603));
    lcd.setCursor(0, 0);
    lcd.print(PPM);
    lcd.print(" PPM de H2");
    break;

/*****MQ-4|CH4*****/
case 12:
    i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
    RS = RL_MQ4 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
    //R0 = 19000;
    rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);
    PPM = pow(10, ((-log10(rs_r0_ratio) + 1.08791) / 0.35816));
    lcd.setCursor(0, 0);
    lcd.print(PPM);
    lcd.print(" PPM de CH4");
    break;

/*****MQ-4|ALCOOL*****/
case 13:
    i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);

```

```

RS = RL_MQ4 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
//R0 = 19000;

rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);

PPM = pow(10, ((-log10(rs_r0_ratio) + 0.78272) / 0.07324));

lcd.setCursor(0, 0);

lcd.print(PPM);

lcd.print(" PPM de ALCOOL");

break;

```

```

/*****MQ-135|CO2*****/

```

case 14:

```

i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);

RS = RL_MQ135 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);

//R0 = 19000;

rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);

PPM = pow(10, ((-log10(rs_r0_ratio) + 0.79349) / 0.38306));

lcd.setCursor(0, 0);

lcd.print(PPM);

lcd.print(" PPM de H2");

break;

```

```

/*****MQ-135|CO*****/

```

case 15:

```

i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);

RS = RL_MQ135 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);

//R0 = 19000;

rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);

PPM = pow(10, ((-log10(rs_r0_ratio) + 0.77915) / 0.28884));

lcd.setCursor(0, 0);

lcd.print(PPM);

lcd.print(" PPM de CH4");

```

```

    break;

/*****MQ-135|NH4*****/

case 16:
    i_sensor_analog_reading = analogRead(SENSOR_ANALOG_PORT);
    RS = RL_MQ135 * ((5 / (VOLT_STEP * i_sensor_analog_reading)) - 1);
    //R0 = 19000;
    rs_r0_ratio = Ajust_ratio((RS / R0), MQ6);
    PPM = pow(10, ((-log10(rs_r0_ratio) + 0.88422) / 0.43706));
    lcd.setCursor(0, 0);
    lcd.print(PPM);
    lcd.print(" PPM de ALCOOL");
    break;

default:
    lcd.setCursor(0, 0);
    lcd.print("SENSOR NAO ");
    lcd.setCursor(0, 1);
    lcd.print("CONFIGURADO ");
}

Serial.println(i_sensor_analog_reading);
}

/***** Pressed_buton() *****/

* Entrada - Void
* Saída - Um inteiro contendo o valor correspondente ao botão apertado
* Obs - O botão apertado equivale à seguinte tabela:
* 1->Botão direito
* 2->Botão superior
* 3->Botão inferior
* 4->Botão esquerdo
* 5->Botão de seleção

```

```
*****
**/
```

```
int Pressed_buton() //Função que discrimina o botão apertado no display
{
    int buton = 0;
    int reading = 0;
    reading = analogRead (0); //Leitura do valor da porta analógica A0
    if (reading < 40) {
        buton = right; //Botão direito
    }
    else if (reading < 110) {
        buton = up; //Botão superior
    }
    else if (reading < 400) {
        buton = down; //Botão inferior
    }
    else if (reading < 600) {
        buton = left; //Botão esquerdo
    }
    else if (reading < 800) {
        buton = select; //Botão de seleção
    }
    return buton;
}
```

```
/****** Ajust_ratio(int r0_rs_ratio, int sensor)
******/
```

```
* Entrada - r0_rs_ratio;sensor
```

```
* Saída - ajustado_ratio
```

```
* Obs - Ajusta o valor da razão r0/rs de acordo com a temperatura e umidade
```

* de acordo com as tabelas fornecidas no datasheet

```

*****
**/

float Ajust_ratio(int r0_rs_ratio, int sensor)
{

    float adjusted_ratio = 0;           //Valor da razao ajustada pela temperatura e
    humidade

    float max_value_ratio = 0;          //Valor de correcao da razao a 33% de
    umidade

    float min_value_ratio = 0;          //Valor de correcao da razao a 85% de
    umidade

    int humidity_ar = dht.readHumidity();           //Leitura da umidade pelo sensor
    int temperature_ar = dht.readTemperature();      //Leitura da temperatura pelo sensor
    if(humidity_ar>85 || humidity_ar<33 || temperature_ar<0 ||temperature_ar>50)
    {
        return NAN;
    }
    delay(1000);
    switch (sensor)
    {
        case MQ6: // SENSOR MQ-6

            max_value_ratio = 0.00008 * pow(temperature_ar, 2) - 0.0067 * temperature_ar + 1.1024;
            //Curva para 33% de umidade

            min_value_ratio = 0.00006 * pow(temperature_ar, 2) - 0.0057 * temperature_ar + 0.9941;
            //Curva para 85% de umidade

            adjusted_ratio = r0_rs_ratio * (max_value_ratio + (min_value_ratio - max_value_ratio) *
            (humidity_ar - 33) / 52); //Interpolacao entre os valores maximos e minimos e a umidade

            break;

        case MQ8: // SENSOR MQ-8

            max_value_ratio = 0.00012 * pow(temperature_ar, 2) - 0.00656 * temperature_ar +
            0.98828; //Curva para 33% de umidade

```

```

    min_value_ratio = 0.00014 * pow(temperature_ar, 2) - 0.00753 * temperature_ar +
1.04125; //Curva para 85% de umidade

    ajusted_ratio = r0_rs_ratio * (max_value_ratio + (min_value_ratio - max_value_ratio) *
(humidity_ar - 33) / 52); //Interpolacao entre os valores maximos e minimos e a umidade

    break;

case MQ4: // SENSOR MQ-4

    max_value_ratio = - 0.0135 * temperature_ar + 1.5181; //Curva para 30% de umidade

    min_value_ratio = - 0.01008 * temperature_ar + 1.13018; //Curva para 85% de umidade

    ajusted_ratio = r0_rs_ratio * (max_value_ratio + (min_value_ratio - max_value_ratio) *
(humidity_ar - 30) / 55); //Interpolacao entre os valores maximos e minimos e a umidade

    break;

case MQ135: // SENSOR MQ-135

    max_value_ratio = -0.000005 * pow(temperature_ar, 3) + 0.000653 *
pow(temperature_ar, 2) - 0.028508 * temperature_ar + 1.366829; //Curva para 33% de
umidade

    min_value_ratio = -0.0000046 * pow(temperature_ar, 3) + 0.000573 *
pow(temperature_ar, 2) - 0.025026 * temperature_ar + 1.2384997; //Curva para 85% de
umidade

    ajusted_ratio = r0_rs_ratio * (max_value_ratio + (min_value_ratio - max_value_ratio) *
(humidity_ar - 33) / 52); //Interpolacao entre os valores maximos e minimos e a umidade

    break;

}

return ajusted_ratio;

}

```

```

/***** Calibration(int sensor)

```

```

*****

```

```

*   Entrada - sensor

```

```

*   Saída   - R0_calibrated

```

```

*   Obs     - Calcula o valor de R0. Assume-se que o sensor está sendo calibrado em um

```

```

*             ambiente com ar limpo

```

```

*****

```

```

**/

```

```

float Calibration(int sensor)
{
    float R0_calibrated = 0;                                //Valor de R0 apos a calibração
    float sensor_resistence_clean_air = 0;                  //Valor da resistência do sensor
    ao ar limpo
    int analog_read = 0;                                     //Valor da leitura da porta analógica
    ligada ao sensor

    switch (sensor)
    {
        case MQ6://sensor MQ-6

            for (int i = 0; i < NUMBER_OF_CALIBRATION_SAMPLE; i++)
            {
                analog_read = analogRead(SENSOR_ANALOG_PORT);
                sensor_resistence_clean_air += RL_MQ6 * ((5 / (0.004883 * analog_read)) - 1);
                delay(CALIBRATION_SAMPLING_TIME);

                lcd.setCursor(0, 1);                          //Permite mostrar o andamento da
                calibração ao usuário

                lcd.print((i * 100 / NUMBER_OF_CALIBRATION_SAMPLE)); //Permite
                mostrar o andamento da calibração ao usuário

                lcd.print("%");                                //Permite mostrar o andamento da
                calibração ao usuário
            }

            R0_calibrated = sensor_resistence_clean_air / (Ajust_ratio(9.8, sensor) *
            NUMBER_OF_CALIBRATION_SAMPLE); //9.8 é o padrão para ar limpo

            return R0_calibrated;

            break;

        case MQ8://sensor MQ-8

            for (int i = 0; i < NUMBER_OF_CALIBRATION_SAMPLE; i++)

```

```

{
    analog_read = analogRead(SENSOR_ANALOG_PORT);
    sensor_resistence_clean_air += RL_MQ8 * ((5 / (0.004883 * analog_read)) - 1);
    delay(CALIBRATION_SAMPLING_TIME);

    lcd.setCursor(0, 1); //Permite mostrar o andamento da
    calibração ao usuário

    lcd.print((i * 100 / NUMBER_OF_CALIBRATION_SAMPLE)); //Permite
    mostrar o andamento da calibração ao usuário

    lcd.print("%"); //Permite mostrar o andamento da
    calibração ao usuário
}

R0_calibrated = sensor_resistence_clean_air / (Ajust_ratio(71, sensor) *
NUMBER_OF_CALIBRATION_SAMPLE); //71 é o padrão para ar limpo

return R0_calibrated;

break;

case MQ4://sensor MQ-4

for (int i = 0; i < NUMBER_OF_CALIBRATION_SAMPLE; i++)
{
    analog_read = analogRead(SENSOR_ANALOG_PORT);
    sensor_resistence_clean_air += RL_MQ4 * ((5 / (0.004883 * analog_read)) - 1);
    delay(CALIBRATION_SAMPLING_TIME);

    lcd.setCursor(0, 1); //Permite mostrar o andamento da
    calibração ao usuário

    lcd.print((i * 100 / NUMBER_OF_CALIBRATION_SAMPLE)); //Permite
    mostrar o andamento da calibração ao usuário

    lcd.print("%"); //Permite mostrar o andamento da
    calibração ao usuário
}

R0_calibrated = sensor_resistence_clean_air / (Ajust_ratio(4.44, sensor) *
NUMBER_OF_CALIBRATION_SAMPLE); //4.44 é o padrão para ar limpo

```

```

    return R0_calibrated;

    break;

case MQ135://sensor MQ-135

    for (int i = 0; i < NUMBER_OF_CALIBRATION_SAMPLE; i++)
    {
        analog_read = analogRead(SENSOR_ANALOG_PORT);

        sensor_resistence_clean_air += RL_MQ135 * ((5 / (0.004883 * analog_read)) - 1);

        delay(CALIBRATION_SAMPLING_TIME);

        lcd.setCursor(0, 1);                                     //Permite mostrar o andamento da
        calibração ao usuário

        lcd.print((i * 100 / NUMBER_OF_CALIBRATION_SAMPLE));    //Permite
        mostrar o andamento da calibração ao usuário

        lcd.print("%");                                           //Permite mostrar o andamento da
        calibração ao usuário

    }

    R0_calibrated = sensor_resistence_clean_air / (Ajust_ratio(3.67, sensor) *
    NUMBER_OF_CALIBRATION_SAMPLE); //3.67 é o padrão para ar limpo

    return R0_calibrated;

    break;

}

}

/***** Configuration_Menu()
*****/

* Entrada - Void

* Saída - Coloca em uma variável global (i_selected_config) o valor correspondente a uma
determinada configuração

* Obs - A função configura qual sensor e gás será usado. A calibração e aquecimento dos
sensores.

```

* O inteiro de saída corresponde a seguinte tabela:

*

**/

```
void Configuration_Menu()  {
    int i_menu_index = 0;    //Index do menu principal de escolhas
    int i_sub_menu_index = 0; //Index dos sub_menus de escolhas
    int i_buton_pressed = 0; //Armazena o valor do botão pressionado
    int i_selected_gas = -1;  //Armazena o valor do gas selecionado
    int i_selected_sensor = -1; //Armazena o valor do sensor selecionado
    int time_counter = 0;     //Armazena o contador de tempo do aquecedor

    bool exit = true;

    while (exit)
    {
        i_buton_pressed = Pressed_buton();
        switch (i_buton_pressed)
        {
            case right:
                i_sub_menu_index = i_sub_menu_index + 1;
                delay(500);
                if (i_sub_menu_index > sub_menu_index_max_value)
                {
                    i_sub_menu_index = sub_menu_index_max_value;
                }
                break;
            case up:
                i_menu_index = i_menu_index - 1;
```

```
    delay(500);
    i_sub_menu_index = 0;
    if (i_menu_index < 0)
    {
        i_menu_index = 0;
    }
    break;
case down:
    i_menu_index = i_menu_index + 1;
    delay(500);
    i_sub_menu_index = 0;
    if (i_menu_index > menu_index_max_value)
    {
        i_menu_index = menu_index_max_value;
    }
    break;
case left:
    i_sub_menu_index = i_sub_menu_index - 1;
    delay(500);
    if (i_sub_menu_index < 0)
    {
        i_sub_menu_index = 0;
    }
    break;
case select:
    break;
}

switch (i_menu_index)
{
```

```

case 0:

    lcd.setCursor(0, 0);

    lcd.print("Escolha o Sensor");

    switch (i_sub_menu_index)
    {
        case 0:

            lcd.setCursor(0, 1);

            lcd.print("MQ-6 |   ");

            i_selected_sensor = MQ6;

            R0 = R_STANDARD_MQ6;

            break;

        case 1:

            lcd.setCursor(0, 1);

            lcd.print("MQ-8 |   ");

            R0 = R_STANDARD_MQ8;

            i_selected_sensor = MQ8;

            break;

        case 2:

            lcd.setCursor(0, 1);

            lcd.print("MQ-4 |   ");

            R0 = R_STANDARD_MQ4;

            i_selected_sensor = MQ4;

            break;

        case 3:

            lcd.setCursor(0, 1);

            lcd.print("MQ-135 |   ");

            R0 = R_STANDARD_MQ135;

            i_selected_sensor = MQ135;

            break;

        default:;
    }

```

```

    }
    break;
case 1:
    lcd.setCursor(0, 0);
    lcd.print("Escolha o Gas ");
    lcd.setCursor(8, 1);
    switch (i_sub_menu_index)
    {
        case 0:
            if (i_selected_sensor == MQ6 || i_selected_sensor == MQ8 || i_selected_sensor ==
MQ4)
            {
                lcd.print(" LPG ");
                i_selected_gas = LPG;
            }
            else
            {
                i_sub_menu_index++;
            }
            break;
        case 1:
            if (i_selected_sensor == MQ6 || i_selected_sensor == MQ8 || i_selected_sensor ==
MQ4)
            {
                lcd.print(" ALCOOL ");
                i_selected_gas = ALCOOL;
            }
            else
            {
                i_sub_menu_index++;
            }

```

```

        break;
    case 2:
        if (i_selected_sensor == MQ6 || i_selected_sensor == MQ8 || i_selected_sensor ==
MQ4)
        {
            lcd.print(" H2  ");
            i_selected_gas = H2;
        }
        else
        {
            i_sub_menu_index++;
        }
        break;
    case 3:
        if (i_selected_sensor == MQ6 || i_selected_sensor == MQ8 || i_selected_sensor ==
MQ4)
        {
            lcd.print(" CH4  ");
            i_selected_gas = CH4;
        }
        else
        {
            i_sub_menu_index++;
        }
        break;
    case 4:
        if (i_selected_sensor == MQ6 || i_selected_sensor == MQ8 || i_selected_sensor ==
MQ135)
        {
            lcd.print(" CO  ");
            i_selected_gas = CO;
        }

```

```
else
{
    i_sub_menu_index++;
}
break;
case 5:
    if (i_selected_sensor == MQ135)
    {
        lcd.print(" CO2  ");
        i_selected_gas = CO2;
    }
    else
    {
        i_sub_menu_index++;
    }
    break;
case 6:
    if (i_selected_sensor == MQ135)
    {
        lcd.print(" NH4  ");
        i_selected_gas = NH4;
    }
    else
    {
        i_sub_menu_index++;
    }
    break;
default: i_sub_menu_index = 0;
}
break;
```

case 2:

```

lcd.setCursor(0, 0);
lcd.print("Deseja Aquecer? ");
lcd.setCursor(0, 1);
switch (i_sub_menu_index)
{
  case 0:
    lcd.print("Sim      ");
    if (i_buton_pressed == select)
    {
      delay(500);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Aquecendo|Faltam");
      for (time_counter = 0; time_counter < HEAT_TIME; time_counter++)
      {
        lcd.setCursor(0, 1);
        lcd.print(((86400 - time_counter) / 3600));
        lcd.print(" h ");
        lcd.print((((86400 - time_counter) / 60) - (((86400 - time_counter) / 3600) * 60)));
        lcd.print(" min");
        delay(1000);
        i_buton_pressed = Pressed_buton();
        if (i_buton_pressed == select)
        {
          break;
        }
      }
      i_sub_menu_index = 0;
      i_menu_index = 3;
    }
  }
}

```

```

    }
    break;
case 1:
    lcd.print("Nao      ");
    if (i_buton_pressed == select)
    {
        i_sub_menu_index = 0;
        i_menu_index = 3;
    }
    break;
default:
    i_sub_menu_index = 0;
}
break;
case 3:
    lcd.setCursor(0, 0);
    lcd.print("Deseja Calibrar?");
    lcd.setCursor(0, 1);
    Serial.println(i_buton_pressed);
    delay(100);
    i_buton_pressed = Pressed_buton();
    delay(100);
    switch (i_sub_menu_index)
    {
        case 0:
            lcd.print("Sim      ");
            if (i_buton_pressed == select)
            {
                lcd.clear();
                lcd.setCursor(0, 0);

```

```

    lcd.print("Calibrando   ");
    Calibration(i_selected_sensor);
    delay(500);
    lcd.clear();
    lcd.print("Sensor Calibrado");
    lcd.setCursor(0,1);
    lcd.print("R0=");
    lcd.setCursor(3,1);
    lcd.print(R0);
    delay(3000);
    exit = false;
    i_sub_menu_index = 0;
}
break;
case 1:
    lcd.print("Nao       ");
    if (i_buton_pressed == select)
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Usando Padrao  ");
        lcd.setCursor(0, 1);
        lcd.print(R0);
        delay(2000);
        exit = false;
        i_sub_menu_index = 0;           //Zera o index de submenu
    }
    break;
default:
    i_sub_menu_index = 0;

```

```

    }
    break;
default:;
}

if (i_buton_pressed == select && i_selected_sensor != -1 && i_selected_gas != -1 &&
i_menu_index < 3)
{
    i_menu_index = 2;
    i_sub_menu_index = 0;
    delay(500);
    //exit = false;
}
}
switch (i_selected_sensor)
{
case MQ6:                //SENSOR MQ6
    switch (i_selected_gas)
    {
        case LPG:            //LGP
            i_selected_config = 0;
            break;
        case H2:              //H2
            i_selected_config = 1;
            break;
        case CO:              //CO
            i_selected_config = 2;
            break;
        case CH4:             //CH4
            i_selected_config = 3;
            break;
        case ALCOOL:          //ALCOOL

```

```

    i_selected_config = 4;
    break;
default::
}
break;
case MQ8:           //SENSOR MQ8
    switch (i_selected_gas)
    {
        case LPG:           //LGP
            i_selected_config = 5;
            break;
        case H2:           //H2
            i_selected_config = 6;
            break;
        case CO:           //CO
            i_selected_config = 7;
            break;
        case CH4:           //CH4
            i_selected_config = 8;
            break;
        case ALCOOL:           //ALCOOL
            i_selected_config = 9;
            break;
        default::
    }
    break;
case MQ4:           //SENSOR MQ4
    switch (i_selected_gas)
    {
        case LPG:           //LGP

```

```

    i_selected_config = 10;
    break;
case H2:                //H2
    i_selected_config = 11;
    break;
case CH4:                //CH4
    i_selected_config = 12;
    break;
case ALCOOL:            //ALCOOL
    i_selected_config = 13;
    break;
default::;
}
break;
case MQ135:              //SENSOR MQ135
    switch (i_selected_gas)
    {
        case CO2:        //CO2
            i_selected_config = 14;
            break;
        case CO:          //CO
            i_selected_config = 15;
            break;
        case NH4:         //NH4
            i_selected_config = 16;
            break;
        default::;
    }
    break;
default::; } }

```

ANEXO A – Datasheet do sensor MQ-4

TECHNICAL DATA

MQ-4 GAS SENSOR

FEATURES

- * High sensitivity to CH₄, Natural gas.
- * Small sensitivity to alcohol, smoke.
- * Fast response . * Stable and long life * Simple drive circuit

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of CH₄, Natural gas, LNG, avoid the noise of alcohol and cooking fumes and cigarette smoke.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V _c	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
P _L	Load resistance	20K Ω	
R _H	Heater resistance	33 Ω ± 5%	Room Tem
P _H	Heating consumption	less than 750mw	

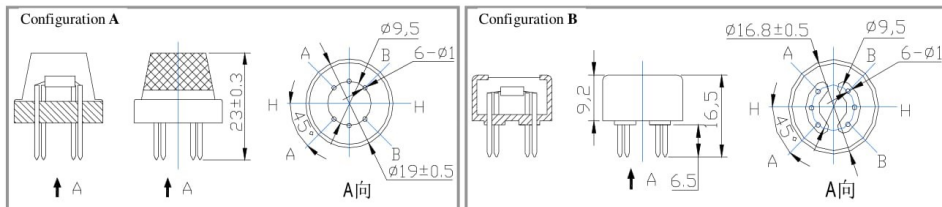
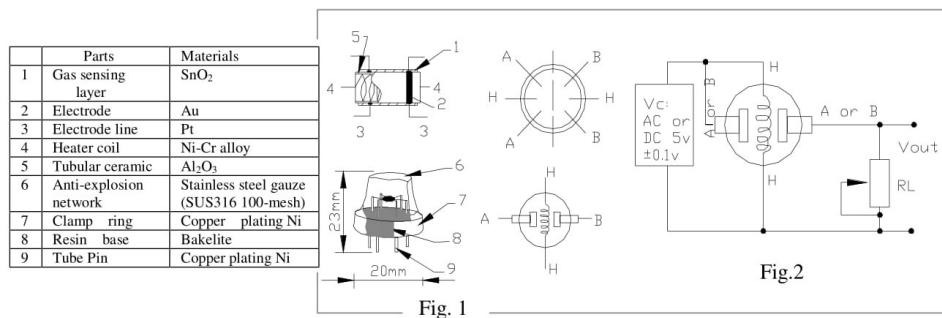
B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-10℃-50℃	
Tas	Storage Tem	-20℃-70℃	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Ramark 2
Rs	Sensing Resistance	10K Ω - 60K Ω (1000ppm CH ₄)	Detecting concentration scope: 200-10000ppm CH ₄ , natural gas
α (1000ppm/ 5000ppm CH ₄)	Concentration slope rate	≤0.6	
Standard detecting condition	Temp: 20℃±2℃ Humidity: 65%±5%	Vc:5V±0.1 Vh: 5V±0.1	
Preheat time	Over 24 hour		

D. Strucyure and configuration, basic measuring circuit



Structure and configuration of MQ-4 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-4 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

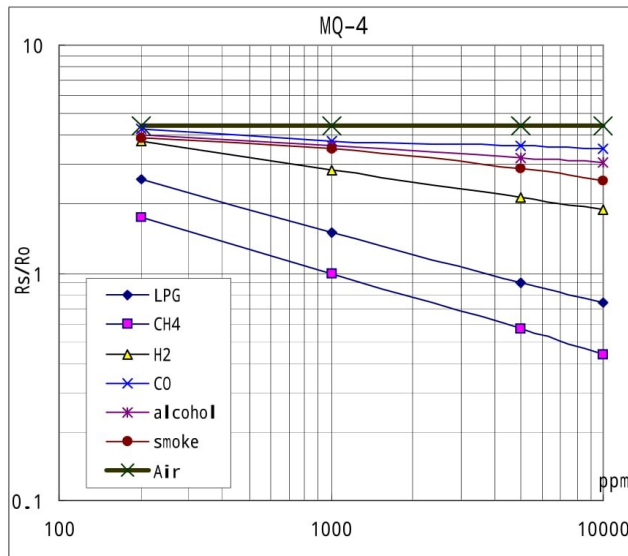


Fig.2 sensitivity characteristics of the MQ-4

Fig.3 is shows the typical sensitivity characteristics of the MQ-4 for several gases.

in their: Temp: 20°C,

Humidity: 65%,

O₂ concentration 21%

RL=20k Ω

Ro: sensor resistance at 1000ppm of CH₄ in the clean air.

Rs:sensor resistance at various concentrations of gases.

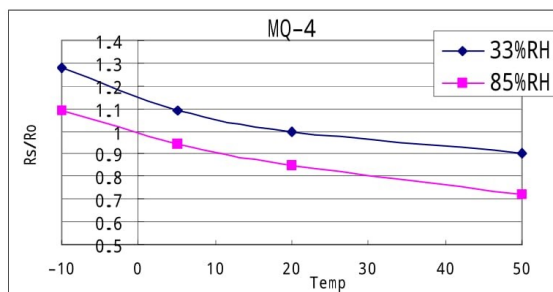


Fig.4 is shows the typical dependence of the MQ-4 on temperature and humidity.

Ro: sensor resistance at 1000ppm of CH₄ in air at 33%RH and 20 degree.

Rs: sensor resistance at 1000ppm of CH₄ in air at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-4 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 5000ppm of CH₄ concentration in air and use value of Load resistance (R_L) about 20K Ω (10K Ω to 47K Ω).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

ANEXO B – Datasheet do sensor MQ-6

TECHNICAL DATA

MQ-6 GAS SENSOR

FEATURES

- * High sensitivity to LPG, iso-butane, propane
- * Small sensitivity to alcohol, smoke.
- * Fast response . * Stable and long life * Simple drive circuit

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of LPG, iso-butane, propane, LNG, avoid the noise of alcohol and cooking fumes and cigarette smoke.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V _c	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
P _L	Load resistance	20K Ω	
R _H	Heater resistance	33 Ω ± 5%	Room Tem
P _H	Heating consumption	less than 750mw	

B. Environment condition

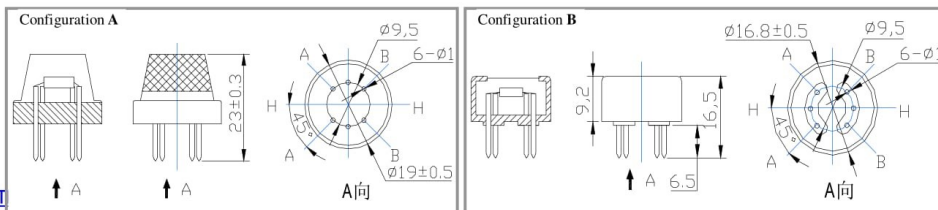
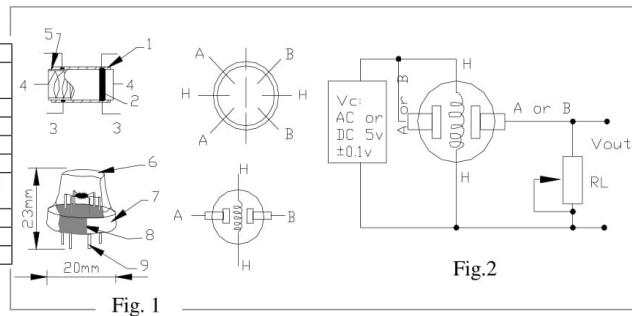
Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-10℃-50℃	
Tas	Storage Tem	-20℃-70℃	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remarks
Rs	Sensing Resistance	10K Ω - 60K Ω (1000ppm LPG)	Detecting concentration scope: 200-10000ppm LPG , iso-butane, propane, LNG
α (1000ppm/ 4000ppm LPG)	Concentration slope rate	≤0.6	
Standard detecting condition	Temp: 20℃ ± 2℃ Humidity: 65%±5%	Vc:5V±0.1 Vh: 5V±0.1	
Preheat time	Over 24 hour		

D. Strucyure and configuration, basic measuring circuit

	Parts	Materials
1	Gas sensing layer	SnO ₂
2	Electrode	Au
3	Electrode line	Pt
4	Heater coil	Ni-Cr alloy
5	Tubular ceramic	Al ₂ O ₃
6	Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7	Clamp ring	Copper plating Ni
8	Resin base	Bakelite
9	Tube Pin	Copper plating Ni



Structure and configuration of MQ-6 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-6 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

Fig.2 sensitivity characteristics of the MQ-6

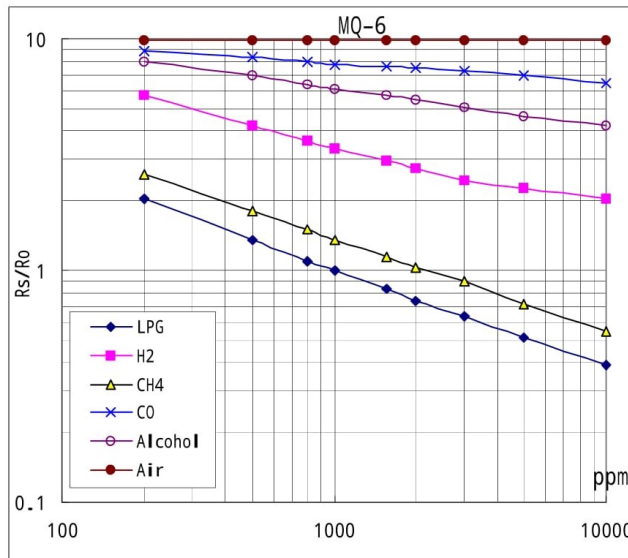


Fig.3 is shows the typical sensitivity characteristics of the MQ-6 for several gases.

in their: Temp: 20°C,
Humidity: 65%,
O₂ concentration 21%
RL=20k Ω

Ro: sensor resistance at 1000ppm of LPG in the clean air.

Rs:sensor resistance at various concentrations of gases.

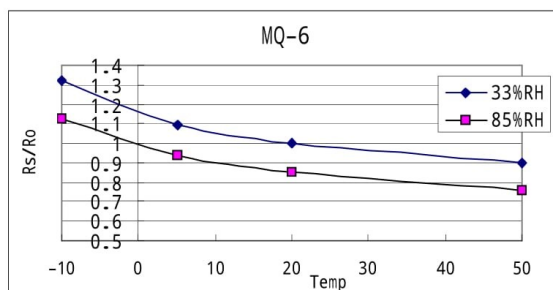


Fig.4 is shows the typical dependence of the MQ-6 on temperature and humidity.

Ro: sensor resistance at 1000ppm of LPG in air at 33%RH and 20 degree.

Rs: sensor resistance at 1000ppm of LPG in air at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-6 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 1000ppm of LPG concentration in air and use value of Load resistance (R_L) about 20K Ω (10K Ω to 47K Ω).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

ANEXO C – Datasheet do sensor MQ-8

TECHNICAL DATA

MQ-8 GAS SENSOR

FEATURES

- * High sensitivity to Hydrogen (H_2)
- * Small sensitivity to alcohol, LPG,cooking fumes
- * Stable and long life

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of Hydrogen (H_2), avoid the noise of alcohol and cooking fumes, LPG,CO.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V_c	Circuit voltage	$5V \pm 0.1$	AC OR DC
V_H	Heating voltage	$5V \pm 0.1$	AC OR DC
P_L	Load resistance	$10K \Omega$	
R_H	Heater resistance	$31 \pm 5\%$	Room Tem
P_H	Heating consumption	less than 800mW	

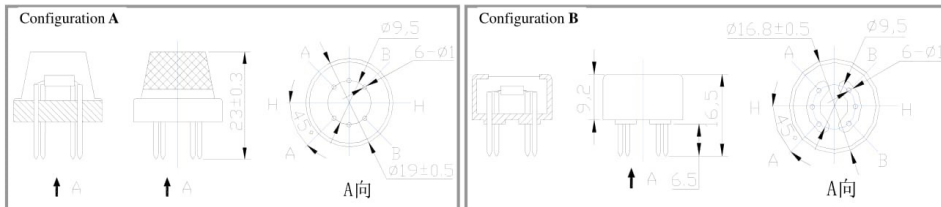
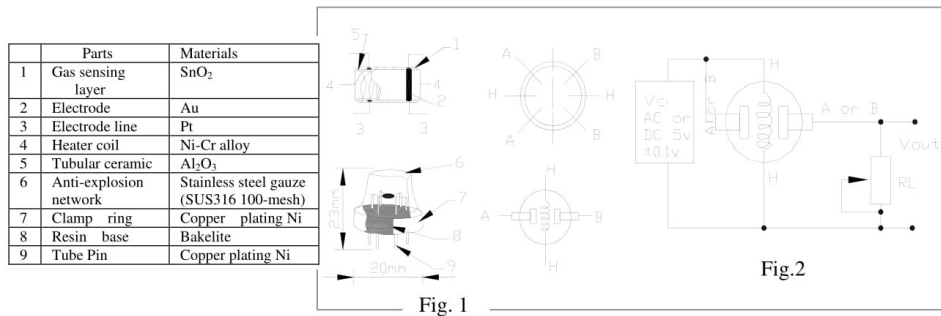
B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
T_{ao}	Using Tem	$-10^\circ C - 50^\circ C$	
T_{as}	Storage Tem	$-20^\circ C - 70^\circ C$	
R_H	Related humidity	less than 95%Rh	
O_2	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Ramark 2
Rs	Sensing Resistance	10K Ω - 60K Ω (1000ppm H ₂)	Detecting concentration scope: 100-10000ppm Hydrogen (H ₂)
α (1000ppm/ 500ppmH ₂)	Concentration slope rate	≤0.6	
Standard detecting condition	Temp: 20℃ ± 2℃ Humidity: 65%±5%	Vc:5V±0.1 Vh: 5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit



Structure and configuration of MQ-8 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-8 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

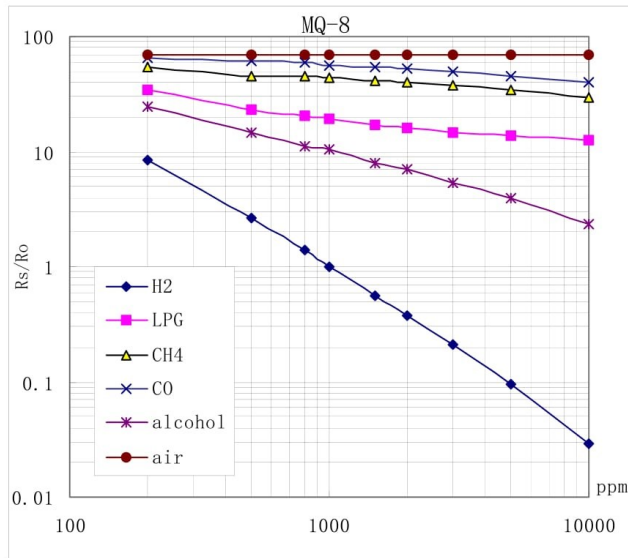


Fig.3 is shows the typical sensitivity characteristics of the MQ-8 for several gases.

in their: Temp: 20°C ,

Humidity: 65% ,

O₂ concentration 21%

RL=10k Ω

Ro: sensor resistance at 1000ppm

H₂ in the clean air.

Rs:sensor resistance at various concentrations of gases.

Fig.2 sensitivity characteristics of the MQ-8

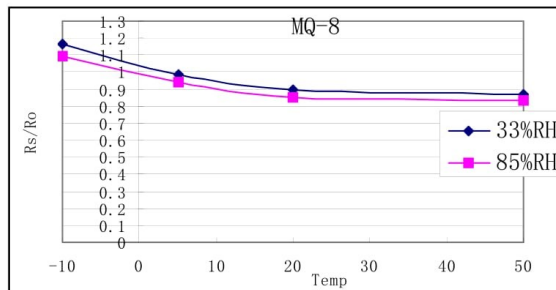


Fig.4 is shows the typical dependence of the MQ-8 on temperature and humidity.

Ro: sensor resistance at 1000ppm of H₂ in air at 33%RH and 20 degree.

Rs: sensor resistance at 1000ppm of H₂ in air at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-8 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 1000ppm H₂ concentration in air and use value of Load resistance (R_L) about 10 K Ω (5K Ω to 33 K Ω).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

ANEXO D – Datasheet do sensor MQ-135

TECHNICAL DATA

MQ-135 GAS SENSOR

FEATURES

Wide detecting scope Fast response and High sensitivity
Stable and long life Simple drive circuit

APPLICATION

They are used in air quality control equipments for buildings/offices, are suitable for detecting of NH₃, NO_x, alcohol, Benzene, smoke, CO₂, etc.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V _c	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
R _L	Load resistance	can adjust	
R _H	Heater resistance	33Ω±5%	Room Tem
P _H	Heating consumption	less than 800mw	

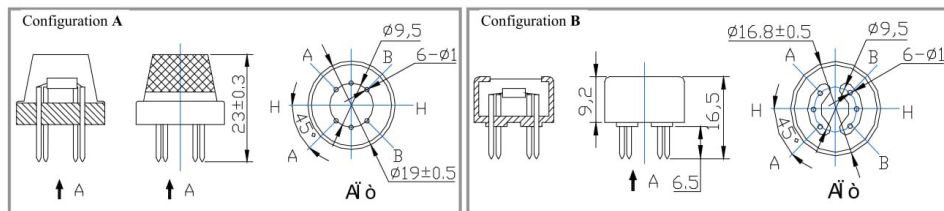
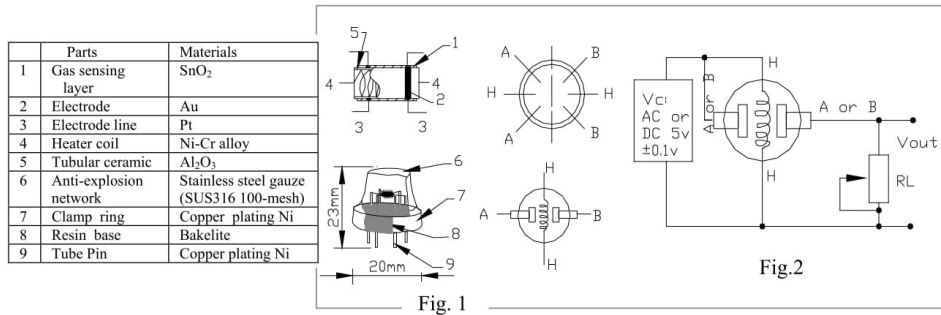
B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-10℃-45℃	minimum value is over 2%
Tas	Storage Tem	-20℃-70℃	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remark 2
Rs	Sensing Resistance	30KΩ-200KΩ (100ppm NH ₃)	Detecting concentration scopell 10ppm-300ppm NH ₃ 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200/50) NH ₃	Concentration Slope rate	≤0.65	
Standard Detecting Condition	Temp: 20±2℃ Vc:5V±0.1 Humidity: 65%±5% Vh: 5V±0.1		
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit



Structure and configuration of MQ-135 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive

components. The enveloped MQ-135 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

Fig.2 sensitivity characteristics of the MQ-135

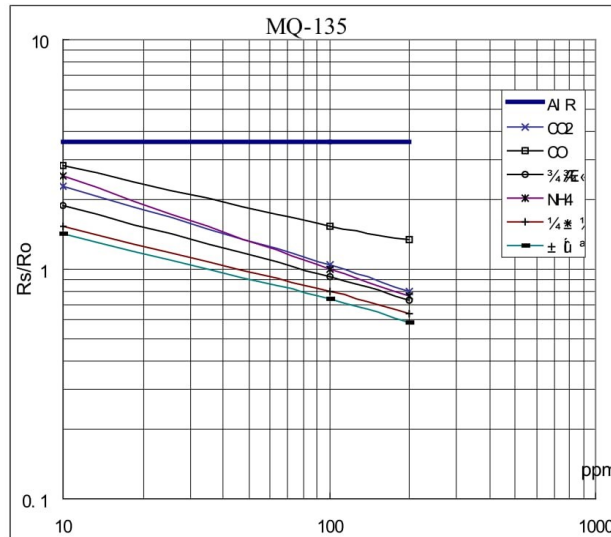


Fig.3 is shows the typical sensitivity characteristics of the MQ-135 for several gases. in their: Temp: 200 °C Humidity: 65% RH O₂ concentration 21% RL=20kΩ

Ro: sensor resistance at 100ppm of NH₃ in the clean air.
Rs: sensor resistance at various concentrations of gases.

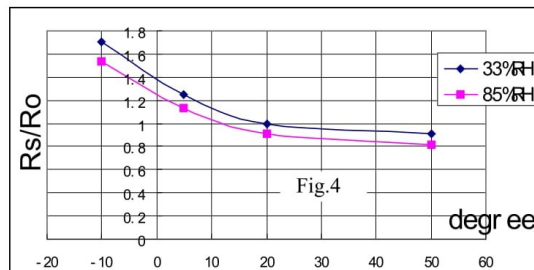


Fig.4 is shows the typical dependence of the MQ-135 on temperature and humidity.

Ro: sensor resistance at 100ppm of NH₃ in air at 33%RH and 20 degree.

Rs: sensor resistance at 100ppm of NH₃ at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-135 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 100ppm NH₃ or 50ppm Alcohol concentration in air and use value of Load resistance that (RL) about 20 kΩ (10kΩ to 47 kΩ).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

