UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ENGENHARIA GEOGRÁFICA, GEOFÍSICA E ENERGIA

Desenvolvimento e teste de um protótipo móvel com sensores low cost

Angelo Roldão Soares

Mestrado Integrado em Engenharia da Energia e do Ambiente

Dissertação orientada por:
Guilherme Carrilho da Graça

2018

Resumo

Apesar de grande parte da poluição ter lugar no exterior, o ambiente interior não está incólume às movimentações de ar que a trazem para dentro dos edifícios, poluição essa nas grandes cidades com origem nos produtos da combustão ou operações industriais. Casos extremos de poluição são, por exemplo, cidades como Beijing com graves casos de *smog* nos últimos anos. A juntar à poluição exterior que inevitavelmente contamina os espaços interiores, temos também a própria habitação do edifício que gera a produção de CO_2 por parte dos ocupantes, o que também afeta a qualidade do ar. Todos estes fatores levam a uma preocupação acrescida na compreensão e monitorização de determinados gases e os seus impactos na saúde humana.

Para se poder afirmar que existem impactos e que existe presença de gases nocivos no ar ambiente, é preciso instrumentação capaz de os detetar. A ideia para esta dissertação surge no contexto da campanha *Breathable Cities World Campaign*, uma campanha filantrópica que visa contribuir para a melhoria da qualidade de vida do homem em completa harmonia com o ambiente.

Com o intuito de quantificar os poluentes, a iniciativa Breathable Cities Campaign utilizou equipamentos extremamente dispendiosos na ordem dos milhares de euros. Esta dissertação procura obter o mesmo tipo de informação quantitativa dos gases, mas através de aparelhos *low cost* na ordem das dezenas de euros utilizando uma placa Arduíno que faz de interface entre uma *cloud network* e os sensores DHT11, MQ-131 e MG-811, servindo assim como plataforma de motivação da dissertação, sendo o principal desafio a construção e teste do protótipo.

Com esta dissertação disponibilizam-se as ferramentas para qualquer académico recrear esta experiência e criar uma estação de monitorização de gases com um peso financeiro menor, o preço de todos os componentes totalizou 158.3 euros e o protótipo tem um peso de cerca de 442 gramas.

O foco na instrumentação foi alvo de grande ponderação, pois é necessário estudar aprofundadamente toda a temática dos sensores e os seus métodos de comunicação, assim como a melhor forma de proceder à construção e teste do aparelho.

Para se testar o funcionamento do protótipo foi prudente considerar dois tipos de calibrações possíveis dos sensores de CO_2 e O_3, porém, o método dos mínimos quadrados revelou-se superior às metodologias de fábrica propostas devido à sua simplicidade. Os dados obtidos foram satisfatórios, no entanto é preciso relembrar que estamos a falar de sensores extremamente low cost e que deverão ser efetuados testes mais extensos em melhores condições de calibração.

Foi possível concluir esta dissertação com um protótipo funcional capaz de fazer medições rudimentares e efetuar comunicação por cloud network. É um *proof-of-concept* com espaço para melhorias como a redução das dimensões e do peso investindo em circuitos integrados ou placas PCB. É possível também testar-se a utilização de sensores mais robustos que podem teoricamente proporcionar melhores resultados de calibração ao depender menos de fatores como a temperatura e a humidade, ou simplesmente assegurar melhores condições de calibração com os sensores que foram testados.

Abstract

Despite a big portion of pollution being generated outdoors, indoor environment is not unaffected by air mass movements that bring that pollution inside. In big cities pollution has it's origins in combustion or industrial operations and can be observed, in extreme, in cities like Beijing, with severe cases of smog in the past years. Besides outdoor pollution that inevitably contaminates indoor environments, we also have the CO_2 production associated with the presence of users in indoors spaces, which by itself also affects air quality. All these factors further the necessity of monitoring and comprehending pollutants and their impacts on human health.

In order to make affirmations about the impact of air pollutants and measure the presence of such pollutants in ambient air, we need instrumentation that is capable of detecting them. The idea for this dissertation comes about in the context of the Breathable Cities World Campaign, a philanthropic campaign whose objective is to promote a better quality of life for mankind and the balance between man and nature.

The Breathable Cities World Campaign, with the purpose of quantifying air pollutants, used extremely expensive equipment with values in the range of thousands of euros. This dissertation procures to obtain the same type of quantifying information, but with low cost devices with a price in the dozens of euros by making use of an Arduino Board, which acts as an interface between a cloud network and a DHT11, MQ-131 and MG-811 sensors. This served as the motivational platform for this dissertation making the construction and testing of a low cost prototype the main focus.

This dissertation gives the tools necessary for any academic to recreate the experience and create a cheap air quality monitoring station. The prototype which weights about 442 grams has amounted to a total expenditure of 158.3 euros.

The focus on instrumentation was target of great analysis as it's necessary to deeply study the whole thematic of sensors and how they operate. It was also important to understand their methods of communication and how we should proceed with the construction and test of the device.

A cautious approach was taken with the calibration of the CO_2 and O_3 sensors taking two different methods into consideration. The minimum squares method revealed itself as the superior approach when compared with factory methodologies, due to it's simplicity. The data gathered from the sensors was satisfactory though it should be noted that we're discussing extremely low cost sensors and we should be careful with taking conclusions without making more extensive tests with better calibration conditions.

It was possible to conclude this dissertation with a functional prototype that was capable of making rudimentary measurements and communicate through cloud networking. It's a proof-of-concept prototype with room for improvement, such as reducing the dimensions and weight by investing in integrated circuits or PCB boards. It's also possible to theoretically obtain better results in calibration tests by testing more robust sensors that have less dependency of factors such as temperature

and humidity, or we could use the same sensors in this dissertation but with better calibration conditions.

Agradecimentos

Gostaria de agradecer a todo o pessoal da Faculdade de Ciências que se disponibilizou para me ajudar. Agradecimentos ao Dimitri por estar sempre disponível. Agradecimentos especiais ao senhor que arruma os carros na minha rua e não deixa a malta dos restaurantes estacionar no meu lugar, pequenas coisas fazem grandes diferenças. Mas os maiores agradecimentos vão para a minha musa, Catarina Ricardo.

Angelo Soares

"You should pray for a healthy mind in a healthy body. Ask for a stout heart that has no fear of death, and deems length of days the least of Nature's gifts that can endure any kind of toil, that knows neither wrath nor desire and thinks the woes and hard labors of Hercules better than the loves and banquets and downy cushions of Sardanapalus. What I commend to you, you can give to yourself; For assuredly, the only road to a life of peace is virtue."

Decimus Iunius Iuvenalis

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

CO_2	Dióxido de carbono
O_3	Ozono
NO_2	Dióxido de nitrogénio
SO_2	Dióxido de enxofre
TiO_2	Dióxido de titânio
V_2O_5	Pentóxido de vanádio
WO_3	Trióxido de tungsténio
CO	Monóxido de carbono
CL_2	Cloro
CH_4	Metano
O^-	Oxigénio
ppm	Partes por milhão
ppb	Partes por bilião
VOC_s	Compostos orgânicos voláteis
EEPROM	Eletrically-Erasable Programmable Read-Only Memory
RISC	Reduced Instruction Set Computer
SRAM	Static Random Access Memory
GPIO	General Purpose Input/Output
I/O	Input/Output
TX	Transmit
RX	Receive
VCC	Voltage common collector
GND	Ground
AT	Attention
EPA	Environmental Protection Agency
NDIR	Nondispersive Infrared
IR	Infrared
UV	Ultraviolet
RTC	Real Time Clock
NTC	Negative Temperature Coefficient
GSM	Global System for Mobile communications
GPRS	General Packet Radio Service
kB/s	kilobytes per second
Mhz	Mega-Hertz
IDE	Integrated Development Environment
A	Ampere
V	Volt
PCB	Printed Circuit Board

SD	Secure Digital
nA	Nanoampere
SPI	Serial Peripheral Interface
I_2C	Inter-Integrated Circuit
UART	Universal Asynchronous Receiver-Transmitter
A/D	Analog-to-Digital
FTP	File Transfer Protocol
APN	Access Point Name
PIN	Personal Identification Number
CHAR	Character (Code)

Capítulo 1

Introdução

Serve o presente capítulo para elucidar o leitor quanto às motivações e intenções desta dissertação.

1.1 Motivação

O primeiro registo de poluição do ar em larga escala ocorreu no ano de 1952, durante a revolução industrial, naquele que ficou conhecido como "*The great smog of 1952*", a crescente utilização do carvão em centros urbanos, neste caso Londres, levou à morte de 12 mil londrinos[1]. Este acontecimento pode ser atribuído à ignorância e falta de monitorização das emissões. Pode aparentar ser um caso isolado do passado, mas não o é, apesar dos avanços na ciência e no pensamento do homem face aos problemas ambientais, a humanidade vive num padrão que não quer abandonar, o modernismo, a qualidade de vida e todas as benesses da ciência na vida do homem têm um preço, esse preço são os produtos de combustão da atividade humana. O processo de combustão é vital a uma grande parte da população mundial e devido ao crescimento contínuo da população e ao progresso económico desenfreado de determinados países, cada vez mais pessoas utilizam meios de transporte como automóveis, autocarros, camiões, etc., especialmente em grandes centros urbanos. Desta premissa nasce a iniciativa *Breathable Cities Campaign*, que visa monitorizar e alertar para o perigo dos gases dos produtos da combustão em meio urbano na saúde humana. O primeiro artigo relacionado com Breathable Cities, foi colocado em prática na Colômbia. O estudo argumenta que segundo os dados da Organização Mundial da Saúde (OMS), 91% da população mundial vive em locais onde a qualidade do ar excede os limites guia da OMS. Esta organização afirma também, que cerca de 4.2 milhões de mortes podem todos os anos ser atribuídas à má qualidade do ar exterior e 3.8 milhões de pessoas morrem em ambientes interiores, vítimas de processos de combustão, como por exemplo fugas de monóxido de carbono[2].

Em relação à poluição e à saúde humana, existe um fio condutor de pensamento, talvez não tão claro, mas profundamente interligado com as palavras proferidas acima, que é a aplicação da Internet das coisas neste tema. Esta dissertação faz uma ponte entre a qualidade do ar e o mundo tecnológico da monitorização da qualidade do ar. Com os crescentes avanços na ciência, é expectável que num futuro onde todos os dispositivos estão interligados e onde qualquer pessoa consegue

ter acesso a aparelhos de monitorização da qualidade do ar *low cost*, irá existir uma maior perceção do impacto individual do ser humano e acreditamos que se irá traduzir numa consciencialização global mais persistente.

1.2 Contextualização

No contexto de Breathable Cities, a correlação entre a exposição a poluentes e os efeitos na saúde estão bem descritos em Guarnieri e Balmes 2014, Loomis et al. 2013 e Kunzli et al 2000[3, 4, 5]. Em Guarnieri e Balmes argumenta-se que o principal mecanismo de ataque destes poluentes na saúde humana, está relacionado com o dano causado nos canais respiratórios devido ao seu poder oxidante, inflamando e tornando a zona mais sensível e suscetível a outros tipos de ataque. Em Loomis et al. 2013 estudou-se o impacto da exposição a partículas em pessoas que vivem ou se movimentam perto de locais onde existe circulação automóvel. Existem fortes indícios que apontam para uma maior probabilidade de contraírem certos tipos de cancro, se frequentarem esses locais regularmente durante pelo menos 72 meses. O artigo de Kunzli et al. 2000 chega a conclusões semelhantes, ao afirmar que a longa exposição à poluição no ar causada por veículos a motor, potencia a mortalidade prematura. Com os impactos na saúde bem documentados, o ponto de foco desta dissertação de mestrado, ao ponderar a iniciativa Breathable Cities, esteve relacionado com a instrumentação, como mencionado na motivação. Qualquer análise gasosa e posteriores alertas da qualidade do ar, partem de uma única fonte, essa fonte é a informação que advém da utilização de instrumentos extremamente dispendiosos. Constituiu interesse, com a motivação dos gases e dos seus efeitos na saúde, proporcionar um método extremamente low cost, que permita a quase qualquer individuo, mas principalmente a um aluno académico, criar uma estação de monitorização de gases low cost.

1.3 Objetivos

Os objetivos desta dissertação eram amplos, passaram por diversas iterações à medida que se foi progredindo na construção do protótipo e chegou-se à conclusão que seria interessante fazer uma documentação aprofundada da construção de um protótipo low cost e fazer uma análise preliminar da performance de alguns sensores comummente utilizados neste tipo de aplicações.

1.4 Estrutura da Tese

Esta dissertação está dividida em seis capítulos que abordam desde a motivação até à construção e teste de um protótipo de monitorização da qualidade do ar ambiente. Serve esta secção para alertar o leitor que as figuras e tabelas estão numeradas por subsecção.

Capítulo 2

Conceitos Teóricos

Nesta secção serão apresentados os conceitos teóricos por de trás dos sistemas de sensores e do funcionamento de todos os dispositivos afetos à dissertação.

2.1 Sistemas de Sensores

Um sistema de sensores é um dispositivo que integra um ou vários sensores, entre outros componentes, necessários para criar um sistema de deteção autónomo e funcional. Existem diversos estudos, principalmente nos últimos anos, que exploram as potencialidades dos sensores low cost.

2.1.1 Dispositivos de monitorização da Qualidade do Ar

No estudo de M.I. Mead et. al[6] é testado um dispositivo composto por três sensores eletroquímicos da empresa Alphasense, sensores de CO, NO, NO_2 que são incorporados num sistema de sensores com comunicação por GPRS e posteriormente replicados e espalhados na área de Cambridge, o estudo tem como objetivo calibrar e testar a fiabilidade em ambiente real. O estudo conclui que os sensores, assumindo que quem vai recrear a experiência tem acesso a condições laboratoriais de calibração de alta qualidade, são bons o suficiente para a utilização na monitorização da qualidade do ar urbano.

Em N. Kularatna et. al[7] é descrito um sistema de monitorização de poluentes no ar, utilizando sensores semicondutores que medem CO, NO_2, SO e O_3. O sistema que integra os sensores exige algum conhecimento de microcontroladores para fazer a comunicação entre sensores e o computador. Os sensores foram calibrados utilizando o método de câmara estática padrão[8]. Segundo o estudo foram registados os valores zero e as amplitudes dos sensores, porém, não foi possível controlar as variáveis da temperatura e humidade relativa no local da realização da experiência, no entanto afirmam, que se situaram entre os 20-25^{o}C e 45-50%. Este estudo reconhece e recomenda inúmeras melhorias do seu sistema apontando algumas das falhas mais importantes. Estas falhas estão relacionadas com as flutuações provocadas pela temperatura e humidade relativa, assim como a degradação dos sensores ao serem expostos a silicones voláteis orgânicos, que podem

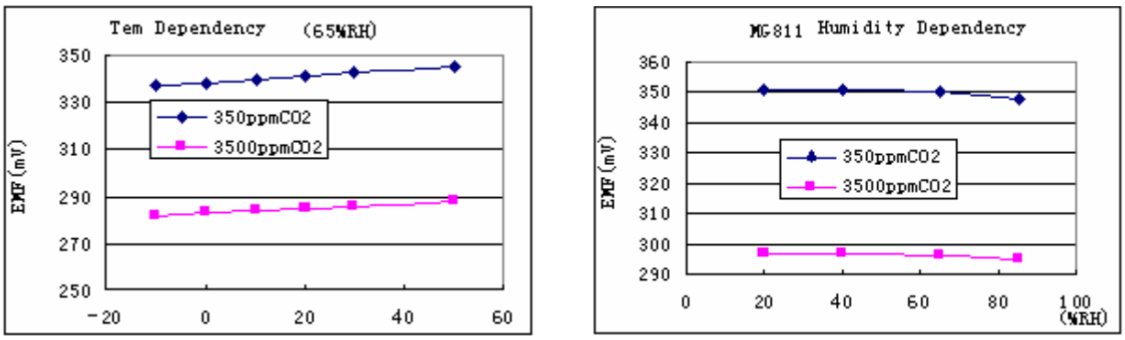
provocar danos irreversíveis, mas afirmam, que apesar de todos os erros que ocorreram, é possível uma boa implementação de um sistema de sensores, assim como é possível utilizá-los com algum grau de certeza para monitorizar as concentrações dos gases no ambiente, neste caso o ambiente no interior de um laboratório.

Por último, assimilamos o estudo de redes de sistemas de sensores[9], onde se utilizaram os sensores MG-811 e o MQ-131. O objetivo foi criar uma rede de sensores wireless de forma a monitorizar a qualidade do ar ambiente. Os métodos de calibração adotados, consistiram em colocar os sensores low cost dentro de uma caixa de plástico, juntamente com o aparelho de alta qualidade de referência, o sensor GrayWolf. Foram injetados na caixa os gases a serem medidos e os valores registados nos sensores foram utilizados para calibrar os sensores low cost face à referência do GrayWolf. A maneira de injetar os gases na caixa de plástico não é especificada e foram utilizados métodos rudimentares queimando papel para produzir CO, CO_2 e outros compostos voláteis. O artigo apresentado parece algo dúbio e não refere a boa precisão dos sensores nas suas conclusões, porém, analisando os dados, parecem existir algumas discrepâncias assim como uma fraca sensibilidade dos sensores testados, o MG-811 e o MQ-131.

2.2 Variáveis em estudo

2.2.1 Temperatura e Humidade Relativa

Estas duas variáveis, no contexto desta dissertação, são duas variáveis de extrema importância no que diz respeito à performance dos sensores. As medições efetuadas pelos sensores dependem das suas resistências internas, que estão diretamente relacionadas com reações físicas e/ou químicas nos sensores. Estas reações, independentes da concentração de gás no ar ambiente, manifestam-se em flutuações na tensão de saída. Nas seguintes figuras retiradas dos *datasheets* do fabricante do MG-811, é possível observar a correlação entre a variação da temperatura e humidade com a tensão de saída:



(a) Influência da Temperatura

(b) Influência da Humidade Relativa

Figura 2.2.1: Influência da temperatura e humidade nos valores de tensão de saída do MG-811

Seguidamente podemos observar os impactos da temperatura e humidade relativa na resistência interna do MQ-131, que se traduz em diferenças na tensão de saída. A seguinte figura é retirada do datasheet do fabricante do sensor em questão.

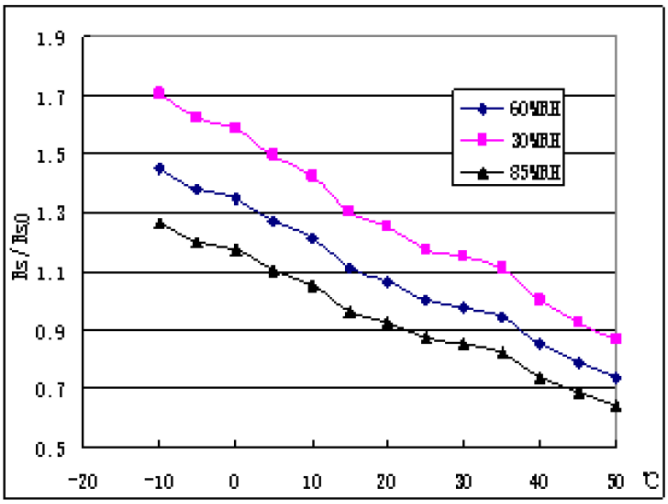


Figura 2.2.2: Influência da temperatura e humidade relativa na relação de resistências do sensor MQ-131

Depreendemos das figuras, que uma variação de 10°C no sensor MG-811, dos 20 aos 30 °C, pode incorrer numa variação de 5 mV. Pode não aparentar ser muito, mas relembramos que isto é uma calibração de fábrica que se assume ser realizada em condições quase perfeitas, sem ruído eletrónico ou variáveis externas, coisa que não é possível garantir nesta dissertação. Por mais pequena que seja a variação em tensão, poderá ser traduzida depois em uma incerteza de 100 a 300 ppm como iremos observar nos resultados.

2.2.2 Dióxido de Carbono

Com efeitos em processos cognitivos, como em processos de decisão e performance em exames, o gás dióxido de carbono (CO_2) é o gás antropogénico de efeito de estufa principal, assim como um bom indicador da qualidade do ar e da ventilação em ambiente interior[10]. O CO_2 é um dos poluentes com grande impacto nos utilizadores e gestores dos espaços interiores, verificando-se a existência de valores máximos recomendáveis aos quais o ser humano deve ser exposto. A sua concentração atual na atmosfera é de cerca de 407 ppm [11], porém em ambiente interior, devido à utilização do espaço por parte de seres humanos, existe uma maior preocupação na acumulação de CO_2, tanto com origem em processos do sistema respiratório, como através da queima de combustíveis fosseis (esquentadores, aquecedores, etc.). Em espaços interiores existem dois regimes de valores limite aconselháveis, o regime prolongado ou regime instantâneo, em regime prolongado são aconselháveis que os valores de CO_2 se mantenham abaixo dos 5.000 ppm no dia-a-dia de 8 horas de trabalho. Em regime instantâneo, o limite é de cerca de 15000 ppm durante 15 minutos pois pode ser fatal[12]. Esta informação oferece uma base de comparação com os dados que serão recolhidos, porém, estes valores são valores limite que dificilmente se atingem naturalmente num

espaço fechado, no entanto existe um leque de sintomas como sonolência, dores de cabeça ou falta de atenção que se manifestam perto dos 5.000 ppm[13].

2.2.3 Ozono

O chamado Ozono (O_3) de baixa altitude, na zona troposférica, é considerado um poluente atmosférico[14]. Não é emitido diretamente por motores de combustão ou operações industriais, mas é formado por processos fotoquímicos ao reagir com os produtos de combustão, como hidrocarbonetos, óxidos de nitrogénio e VOCs[15]. Segundo o estudo de Delfino, R. J. et. al, existe uma relação, entre os níveis registados de O_3 e a entrada de idosos acima de 64 anos nos hospitais com problemas respiratórios, pois devido à idade, estão mais vulneráveis a este tipo de poluição, afetando especialmente doentes com asma[16]. Segundo o Decreto-Lei n.º 102/2010, de 23 de Setembro, são emitidas as informações ou alertas quando os valores ultrapassam os limites observados na seguinte tabela:

Tabela 2.2.1: Limiares de informação e alerta para o ozono

Objetivo	Período de referência	Limiar
Informação	Uma hora	$180\,\mu g/m^3$
Alerta	Uma hora	$240\,\mu g/m^3$

Estes valores têm de ser registados durante 3 horas consecutivas, em localizações que representem a qualidade do ar numa área mínima de $100\,km^2$ ou na totalidade de uma zona ou aglomeração, consoante a que for menor.

2.3 Organização Mundial da Saúde

A Organização Mundial da Saúde (OMS), é uma organização pertencente à ONU, que tem como uma das suas principais atividades a definição de diretrizes gerais para a condução das políticas públicas internacionais sobre a saúde. Para tal, estabelece recomendações quanto à adoção de normas e padrões, articula e coordena o avanço do conhecimento sobre as causas e os efeitos dos problemas de saúde, fornece suporte técnico para países e monitoriza e acompanha as mudanças das condições de saúde no mundo[2].

Existem normas específicas aconselhadas pela OMS, para valores limite de exposição a vários gases e partículas. Alguns valores das suas diretrizes referentes a gases poluentes podem ser encontrados na tabela 2.3.1.
Os valores de CO_2 não se encontram na tabela, mas a OMS segue as diretrizes da OSHA, esses valores podem ser observados acima na secção do Dióxido de Carbono. Alguns destes gases poluentes podem ser interessantes medir em futuros projetos de estações low cost de monitorização da qualidade do ar ambiente.

Tabela 2.3.1: Valores limite de exposição recomendados a poluentes comuns

Gás poluente	Concentração ppm	mgm3	Tempo de exposição
CO	10.0	10	8 horas médias
NO_2	0.1	0.20	1 hora média
O_3	0.06	0.12	8 horas médias
SO_2	0.06	0.17	24 horas médias

2.4 Princípios de funcionamento do hardware

Desde a montagem do protótipo até às medições, considera-se que a compreensão teórica do funcionamento dos sensores utilizados é fulcral. Explica-se então seguidamente, a teoria operacional dos componentes mais importantes.

2.4.1 Placa Arduíno

Foi utilizado o Arduíno Uno com um microcontrolador Atmega328 16 MHz, pertence ao microcontrolador de 8-bits da família Atmel, com uma arquitetura RISC. Tem 32 KB de memória flash com capacidades *read-while-write*, 1KB EEPROM, 2KB SRAM, 23 linhas de I/O (GPIO) e 32 linhas comuns de registo, é programável com UART, 2-wire interface, SPI, 6 canais conversores de 10-bit A/D e tem oscilador interno. As suas capacidades são satisfatórias para ser utilizado como interface entre sensores, computado, FTP e cloud network.

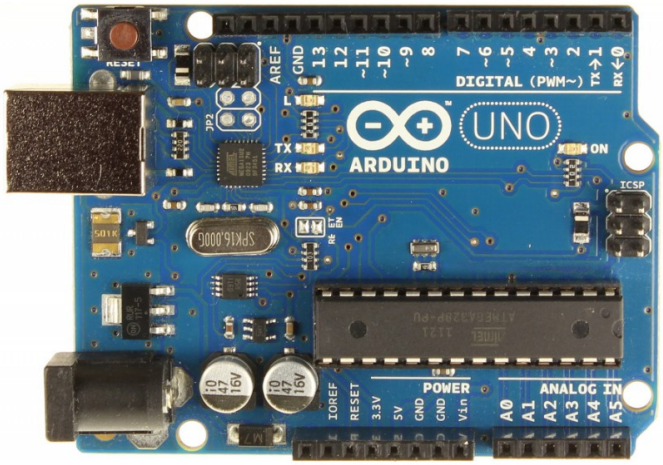


Figura 2.4.1: Imagem de um Arduíno Uno

2.4.2 Sensor amperométrico

Os sensores de gás amperométricos com eletrólitos não aquosos existem desde os anos 70. À medida que a tecnologia foi progredindo, a área de superfície necessária para construir um destes sensores foi sendo reduzida e torna-se possível a construção de sensores low cost de pequenas dimensões. Estes sensores são utilizados para a medição de CL_2, CH_4, CO_2 e CO_2. É comum na literatura, encontrar-se este sensor descrito como polarográfico, eletrólito, voltamétrico, amperométrico ou micro-célula de combustível[17]. Esta nomenclatura pode por vezes induzir o leitor em erro pois este tipo de sensor não é polarográfico, porque não contém um elétrodo de mercúrio, muitos não são eletrólitos, mas sim células galvânicas, porém não geram energia suficiente para ser chamados de micro-célula de combustível. A forma mais correta de apelidar estes sensores de gás é chamá-los de sensores amperométricos de potencial constante.

Um sensor eletroquímico amperométrico é tipicamente constituído por um polímero sólido, elétrodo de trabalho, elétrodo de referência, membrana permeável a gases e contra elétrodo[17]. Um contra elétrodo é um elétrodo secundário que permite, juntamente com o elétrodo de trabalho, a medição de corrente que ocorre num dado processo químico. Neste caso, é utilizado na deteção de uma determinada quantidade de gás no ar, ou seja, à medida que um gás se difunde na célula, determinados gases são reduzidos no cátodo e oxidados no ânodo e é produzida uma diferença de tensão potencial. A corrente elétrica gerada pela reação química é proporcional ao nível de concentração do gás reagente[18].

Inerente aos diversos compostos que constituem este sensor, é possível através da modificação da geometria ou das dimensões dos componentes, manipular a performance analítica do sensor indo ao encontro do pretendido. A sensitividade das observações, a seletividade, o tempo de resposta e a estabilidade do sinal[19], são fatores que podem ser configurados pelo fabricante consoante a futura utilização do sensor por partes dos utilizadores. Por norma, uma das alterações mais importantes é o material e a porosidade da membrana que serve de interface de interação entre o gás e os elétrodos. Na figura 2.4.2 é possível observar uma ilustração esquemática de um sensor de gás amperométrico típico.

Apesar de tudo, este sistema não é perfeito e embora seja possível regular a sua sensitividade e seletividade através dos materiais usados, estes devem a sua precisão a condições perfeitas de funcionamento, com temperaturas e humidades relativas constantes para concentrações do gás no ar conhecidas, no entanto, ao contrário de sensores de alta qualidade, nos sensores utilizados nesta dissertação, não existem métodos de compensação para estas variáveis, o que se traduz em erros de leitura. Outras fontes de erro dos sensores low cost são os falsos alarmes. Devido à forma como este sensor opera é possível que outros gases comuns na atmosfera reajam de formas semelhantes ao gás para o qual o dispositivo foi calibrado, resultando num falso positivo. Tudo depende da qualidade dos materiais, sendo que novas tecnologias com materiais nanocompósitos de nanotubos de carbono asseguram melhores medições[20]. Estes sensores têm um tempo de vida limitado e vão-se desgastando com o passar do tempo, onde a sua taxa de desgaste está relacionada

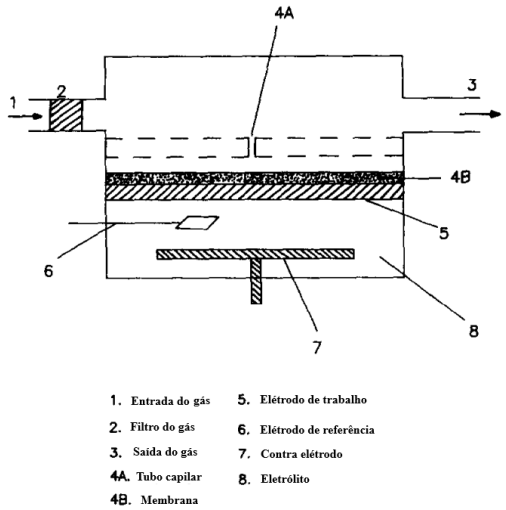


Figura 2.4.2: Representação esquemática de um sensor amperométrico típico

com a qualidade do material e com a exposição do sensor aos gases reagentes. A taxa de desgaste depende também da tensão aplicada que passa pela membrana, é necessário manter os valores recomendados pelo fabricante ou podemos danificar o material e modificar as suas propriedades, a membrana pode também ser danificada por excesso de humidade ou poeiras no ambiente que se pretende monitorizar. Decidir quando é altura de recalibrar estes sensores de modo a manter uma precisão desejada, pode revelar-se um problema e está relacionado com o seu drift. O drift do sensor é o quanto este tende a desviar-se da calibração efetuada e quanto tempo demora a obter esse desvio, daí a dificuldade em operar com sensores low cost onde, por norma, o seu drift é muito maior que um sensor de alta qualidade[18].

Na figura 2.4.3 temos um exemplo de um sensor eletroquímico da WINSEN.

Figura 2.4.3: Imagem do sensor MG-811 da WINSEN

2.4.3 Sensor semicondutor

Os semicondutores de óxido metálico (MOS) são os sensores mais comummente utilizados em variados ramos da indústria devido ao seu relativo baixo custo, à sua robustez, pouco peso, beneficiam de um bom grau de sensitividade e têm baixos tempos de resposta. Existem dois tipos principais de óxidos metálicos, os de transição e não-transição. Os óxidos metálicos de transição têm uma grande variedade de estados de oxidação na superfície do óxido metálico, com configurações eletrónicas dos seus eletrões de valência d0 e d10, isto torna-os bons candidatos a serem usados em aplicações de medições de gases, ao contrário dos de não-transição, que não possuem grande variedade de estados. A configuração d0 pode ser encontrada em óxidos metálicos de transição (e.g. TiO_2, V_2O_5, WO_3), enquanto que a configuração d10 aparece em óxidos metálicos de transição posterior, SnO_2 e ZnO, sendo que o sensor MQ-131 é precisamente um semicondutor de SnO_2[21].

Ainda referente aos tipos de semicondutores, existem, n-type e p-type. A diferença entre os dois aplicados aos sensores reside nas suas amplitudes de temperatura de operação, sendo que os semicondutores p-type operam a temperaturas inferiores[22].

Estes sensores são utilizados maioritariamente para detetar um determinado gás através de reações de redox entre o gás que se quer detetar e a superfície de oxido metálico do sensor. Sabemos através da literatura**??** que o gás ao interagir com a superfície de oxidação do metal e ao interagir com o oxigénio (O-) adsorvido, resulta numa mudança da concentração dos portadores de carga no material. Esta mudança altera a condutividade do material, provocando assim variações de tensão. Nos semicondutores n-type (MQ-131) a maioria dos portadores de carga são os eletrões que ao interagirem com o gás redutor, aumentam a condutividade. De toda a gama de sensores semicondutores baseados em óxidos metálicos, aquele que se tornou mais popular foi o de SnO_2, onde duas das razões para tal são o facto da sua sensitividade ser bastante alta e a sua amplitude de temperaturas de operação ser também ela abrangente, indo desde os 25^{o}C até aos 500^{o}C. Por outras palavras, antes de se efetuar medições, estes sensores são aquecidos através de corrente elétrica, que passa num filamento até atingir uma temperatura de operação de 300-450 ^{o}C, pois é feita a afirmação que esta temperatura de operação é a temperatura ideal, onde o O- é dominante. Porém, faz-se a ressalva que diferentes temperaturas de operação traduzem-se em sensitividades diferentes a gases diferentes, o CO por exemplo é melhor detetado se a temperatura de operação for de 90^{o}C, enquanto que a 400^{o}C se deteta melhor o CH_4[23]. Como igualmente referido nos sensores amperométricos, os sensores mais dispendiosos têm métodos de compensação da influência de temperaturas externas, deste modo previnem grandes flutuações na temperatura de operação, enquanto que os sensores low cost utilizados nesta dissertação não têm essa capacidade, daí a previsão da sua menor fiabilidade. Na figura 2.4.4 temos um diagrama típico de um sensor amperométrico e nas figuras 2.4.6 e 2.4.5 temos exemplos reais desses sensores, versão low cost e high cost respetivamente.

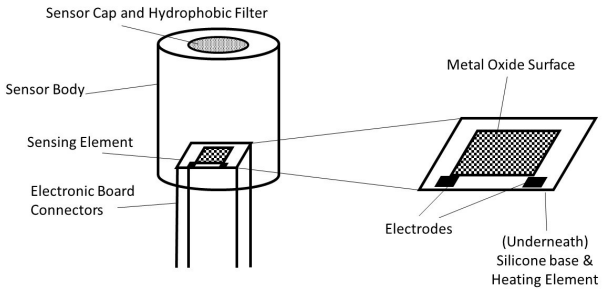


Figura 2.4.4: Sensor típico semicondutor

Figura 2.4.5: Imagem do sensor MQ-131 da WINSEN

Figura 2.4.6: Imagem do sensor Aeroqual 500 Series

2.4.4 Sensor de infravermelhos não dispersivos

Os sensores de infravermelhos não dispersivos NDIR (non-dispersive infrared) pertencem à categoria de sensores óticos. Este tipo de sensor torna possível a deteção de gases com uma sensitividade, seletividade e estabilidade superior a métodos não óticos. Este tipo de sensor tem um tempo de vida superior aos demais e o seu tempo de resposta é relativamente baixo. Este tipo de tecnologia é bastante utilizado para a deteção de poluentes no ar como o CO, SO_2, NO_x, CO_2, entre outros gases. Os sensores NDIR são fiáveis e são particularmente robustos contra a interferência de outros componentes presentes no ar que não o gás que se tenciona medir[24].

Existem diversos tipos de espectroscopia, alguns tipos bastante dispendiosos, mas que conferem grande grau de sensitividade e seletividade. Para efeitos desta dissertação, o sensor IR disponível baseia-se na espectroscopia de absorção básica, mais propriamente, no princípio da espectroscopia de absorção molecular. Este princípio é dependente da concentração do gás, pois depende da absortividade molecular dos fotões a determinados comprimentos de onda. Estes seguem seguem os fundamentos da lei de Beer-Lambert[25]. Por outros termos, cada gás tem uma determinada constante de absorção que por sua vez se traduz em comprimentos de onda diferentes, que reagem de uma forma previsível quando passam por um feixe infravermelho, criando a sua própria impressão digital. Podemos observar na alínea a) da figura 2.4.7 uma ilustração esquemática básica do funcionamento de um sensor NDIR. Na alínea b) da figura 2.4.7 temos uma modificação comum, que adiciona um detetor extra ajudando a eliminar interferência externa sem aumentar excessivamente a complexidade do sistema[23].

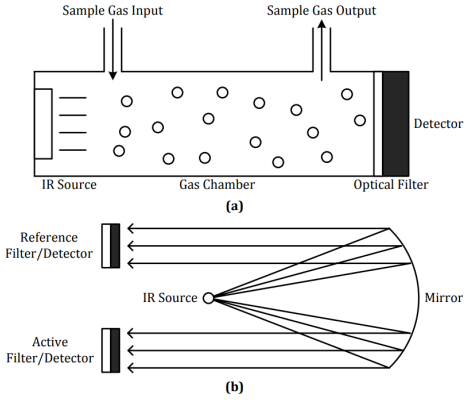


Figura 2.4.7: Sensor típico NDIR

Neste caso o gás a ser medido será o dióxido de carbono através do sensor de referência Telaire 7001.

Figura 2.4.8: Imagem do sensor Telaire e o seu DataLogger

2.4.5 Real Time Clock

Um *Real Time Clock* (RTC), é utilizado no projeto de modo a manter o registo do ano, mês e dia, assim como as horas, minutos e segundos aquando da realização das calibrações. A maneira como o consegue deve-se à utilização de um circuito, cujo objetivo é gerar ondas sinusoidais onde a sua frequência é determinada por um cristal piezoelétrico, normalmente feito de quartzo. O modelo utilizado neste projeto é o ds1307. Segundo alguns entusiastas em fóruns do Arduíno, a qualidade do cristal piezoelétrico deste modelo é de baixa qualidade e é altamente influenciável por flutuações de temperatura, levando a perda de informação no que diz respeito a manter a hora certa. Quando o RTC não está ligado, o registo das horas e data é mantida através de uma pilha CR2032 que pode durar anos, sendo que o circuito só necessita de 500 nA para manter a contagem do tempo.

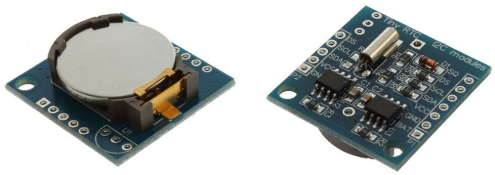


Figura 2.4.9: Imagem do componente RTC ds1307

2.4.6 Sensor de temperatura e humidade

O DHT11 presente na figura 2.4.10 é um sensor de temperatura e humidade com dois componentes distintos. Um deles é um termístor NTC(Negative temperature coefficient), este termístor, geralmente feito de cerâmica ou polímero, tem uma resistência interna que varia consoante a temperatura do ar ambiente. Como estamos a falar de um NTC, a resistência decresce quando a temperatura aumenta.

O segundo componente, que mede a humidade, é constituído por dois elétrodos que seguram uma placa de absorção de humidade no meio e quando existem alterações de humidade neste substrato, existem variações na condutividade, ou por outras palavras, a resistência entre os dois elétrodos muda. Esta mudança é medida e processada por um circuito integrado que a transmite a um microcontrolador para ser interpretada[26].

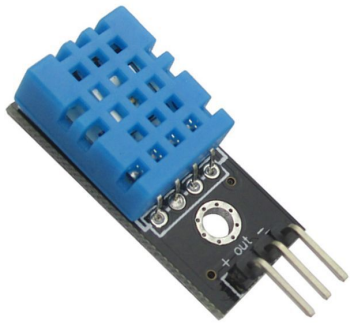


Figura 2.4.10: Imagem do sensor DHT11

2.4.7 Módulo GSM/GPRS

Um modem GSM (Global System for Mobile communications), é um modem wireless que se comporta de uma forma semelhante a um modem dial-up. A principal diferença é que o dial-up envia e recebe informação através de uma linha fixa de telefone, enquanto que um modem wireless envia e recebe informação através de ondas de radio. Embora não se tenham encontrado fontes científicas para a seguinte afirmação, é recorrente dizer-se que um modem GSM tem uma velocidade máxima aproximada de envio e receção de informação de 9.6 kB/s, que é uma velocidade baixa.

Um modem GPRS (General Packet Radio Service) é um modem GSM que tem a adição da tecnologia GPRS, tecnologia essa com um modo de funcionamento semelhante à Internet, onde a informação enviada é dividida em pacotes menores, que serão transmitidos aos destinatários consoante a sua necessidade, sendo reconstruidos como um puzzle ao ser recebidos no destinatário. Esta funcionalidade permite que vários utilizadores estejam ligados à mesma rede, pois é feita uma gestão da distribuição da informação. Com este tipo de funcionamento de transmissão de informação por pacotes, à semelhança da Internet, o GPRS é utilizado de modo a fazer a interoperabilidade entre a Internet e as redes GPRS existentes. É possível aceder a qualquer tipo de serviço

utilizado na Internet como FTP, cloud network, navegação web, chat, email etc., pois está tudo disponível através desta rede móvel. Comparativamente com o GSM que envia informação até 9.6 kB/s, o GPRS é capaz de enviar ou receber informação com uma velocidade máxima teórica de 171 kB/s[27, 28].

Quanto ao modem especificamente utilizado, temos o módulo SIM900A, que é um módulo GSM/GPRS de dupla banda. O modem em questão comunica nas frequências 900 e 1800 MHz através de comandos AT (*Attention*). Este tipo de comandos foram criados por Dennis Hayes e são utilizados na generalidade dos modems.
Na figura 2.4.11 vemos o modelo utilizado assim como os únicos extras necessários para o seu funcionamento, a antena e o cabo molex já vêm incluídos e basta colocá-los nas respetivas entradas. Por último, basta adquirir dois fios jumper fêmea-macho e liga-los no TX e RX para posteriormente serem ligados ao Arduíno.

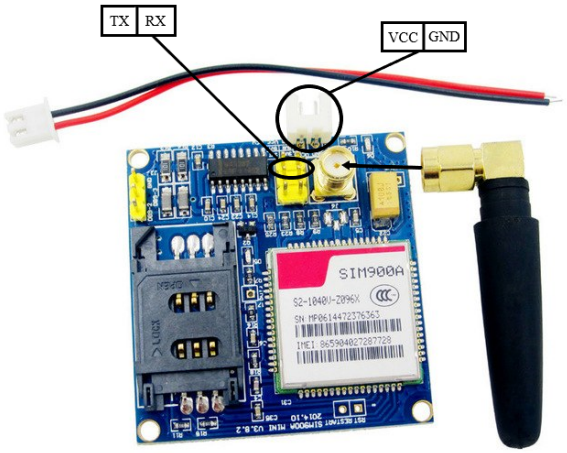


Figura 2.4.11: Módulo Modem SIM900A e respetivas ligações

Capítulo 3

Métodos e Materiais

No presente capitulo serão discutidos os critérios de pesquisa assim como todo os aspetos inerentes ao funcionamento dos componentes e construção do protótipo.

3.1 Critérios de pesquisa

Após a leitura dos artigos referenciados no capítulo acima, tornou-se claro que as informações referentes ao planeamento e construção de um protótipo, com o intuito de monitorizar a qualidade do ar não são as mais detalhadas, os processos não são claros e é necessário normalmente um conhecimento eletrotécnico muito específico. Este facto não é de estranhar dado que em termos académicos parece haver um foco na construção de redes de monitorização ou no teste dos sensores de uma empresa particular e não no protótipo em si. Assim sendo, foi utilizada uma plataforma mais *user-friendly* para se elaborar este projeto, a plataforma Arduíno. A razão para tal deve-se à enormidade de informação disponível na Internet por parte de entusiastas, *bloggers*, participantes de fóruns Arduíno, etc., que discutem entre si, maneiras viáveis de construir as suas próprias estações de monitorização da qualidade do ar. Nesta dissertação, utilizaram-se essas fontes de informação como inspiração. No que diz respeito a códigos de programação, foi amplamente utilizado o repositório *Github* em busca de códigos ou pedaços de código que pudessem ser utilizados ou alterados, de modo a se encaixarem naquilo que seria necessário para este projeto. Todos os códigos utilizados, alterados e criados para garantir o bom funcionamento do protótipo, encontram-se discriminados no Anexo A.1.

As supramencionadas pesquisas de artigos e *websites* elucidaram o autor desta dissertação em diversos aspetos que serão mencionados neste capitulo, como parte da metodologia da escolha dos sensores e como construir o protótipo.

Existem diversos fabricantes ou distribuidores de sensores low cost, porém, esta nomenclatura "low cost" não se encaixa com aquilo que se considera low cost nesta dissertação. Segundo dados da EPA[29], um dispositivo low cost é um dispositivo com um preço inferior a 2500 euros. Em determinados artigos científicos mais conservadores, quando são referidos sensores low cost, a grande maioria está a referir-se a sensores de centenas de euros. Nesta dissertação, quando se

faz referência a sensores low cost, estamos a fazer referência a sensores de dezenas de euros. Daqui em diante descartou-se qualquer investimento em sensores na ordem das centenas de euros e a atenção foi focada em sensores de fabrico chinês, onde muitos deles, têm origem na empresa WINSEN e estão vastamente difundidos em websites como o Ebay, Aliexpress e Amazon.

No caso desta dissertação, sempre que possível e de modo a usufruir das benesses da faculdade de ciências, utilizaram-se duas lojas, a ptrobotics e a mauser. Os sensores escolhidos, assim como todos os materiais referentes à tese, encontram-se na tabela da subsecção seguinte intitulada de Componentes do Protótipo 3.2.1.

A escolha dos componentes foi alvo de escrutínio. É fulcral certificar que todos os componentes são compatíveis com o Arduíno, ou seja, que conseguem comunicar por SPI, UART, I2C ou Analog/Digital. Convém também, para aqueles que usam códigos UART, SPI ou I2C, verificar se existem *libraries* ou códigos disponíveis antes de os encomendar, pois programar de raiz pode ser muito complexo para quem não tiver um nível mediano de conhecimento em C++ para lidar com *byte addressing*.

Apesar de todos os cuidados com as escolhas de componentes, foram escolhidos alguns componentes que a comunidade online tende a afirmar como de qualidade inferior, como é o caso do MG-811 e o RTC ds1307, porém, do ponto de vista desta dissertação, isso só os torna mais interessantes, proporcionando curiosidade em aferir cientificamente a sua fiabilidade e retirar algumas conclusões acerca da sua qualidade.

Como mencionado, embora se tenham feito os esforços para verificar o máximo número de nuances na aquisição de sensores low cost, existiram alguns fatores que não foram tidos em conta até à aquisição dos mesmos. Um exemplo seria o facto de alguns sensores virem montados em módulos, se o fabricante do módulo tiver disponibilizado no datasheet ou no website da empresa códigos exemplo para se trabalhar com o módulo, a ligação pode ser direta e funcionar perfeitamente, como é o caso do MG-811, mas no caso do MQ-131 o módulo não funcionou devidamente o que implicou a utilização de outros métodos que serão explicados mais à frente.

Ainda menos positivo do que códigos ou módulos inutilizáveis, são os próprios datasheets dos sensores que revelam grande falta de informação e, no pior dos cenários, existe muita informação dúbia devido à existência de diversos sensores cópia dos sensores da WINSEN. Nos próprios datasheets da WINSEN, devido à natureza do sensor low cost, existe falta de informação referente a propriedades do sensor, como o drift e range, que seriam interessantes saber. Aconselha-se atenção redobrada e espírito critico quando se leem os datasheets do MG-811 e o MQ-131.

3.2 Componentes do protótipo

Nesta etapa, após a escolha dos componentes, foi altura de começar a compilar a informação acerca dos mesmos. Dividiu-se a informação em duas tabelas. Na primeira tabela 3.2.1, estão listados os componentes e partes, necessários para a construção do protótipo. Na segunda tabela

Tabela 3.2.1: Componentes do protótipo

Componentes primários	**Respetivos componentes extra**	**Preço (€)**
DHT11	3 cabos jumper	3.7
MG-811	N/A	45.9
MQ-131	7 cabos jumper e 1 resistência 100 kΩ	48.8
Arduíno Uno	2 cabos jumper e DC-DC jack	23.8
SD Shield 3.0	N/A	4.7
RTC ds1307	4 cabos jumper	5.9
2x LM2596S	4 cabos jumper e DC-DC jack barrel	5.2
Ventoinha 50x50mm	N/A	6.6
LED	2 cabos jumper e 1 resistência 330 Ω	0.3
GSM/GPRS	2 cabos jumper	7.6
Caixa 159x140x60mm	N/A	5.8
	TOTAL	**158.3**

3.2.2, são fornecidas informações importantes acerca do que se está a medir, as amplitudes e os consumos de corrente elétrica de cada componente.

Devem ser mencionadas algumas informações extra acerca da tabela 3.2.1: Podemos ver alguns N/A (Não aplicável) referentes ao MG-811 e à ventoinha pois estes componentes não precisam de cabos porque já vêm incluídos. O SD shield 3.0 tem apenas de ser encaixado no Arduíno Uno logo também não precisa de cabos. Na mesma tabela não aparecem alguns componentes, não por serem menos importantes do que aqueles que se encontram na tabela, mas porque podem facilmente ser alterados consoante as necessidades, por exemplo, podemos utilizar uma bateria ou um transformador caso seja necessário um protótipo fixo ou móvel. Para se fazer o dimensionamento de uma bateria são disponibilizados os consumos de corrente elétrica totais na tabela 3.2.2 que foram retirados dos respetivos datasheets consoante alimentação a 5 ou 6 V. Para o protótipo desta dissertação utilizou-se um transformador de 12 V e 2 A que cobre as necessidades do protótipo.. Na seguinte figura podemos observar alguns dos componentes mais importantes prontos a serem testados individualmente.

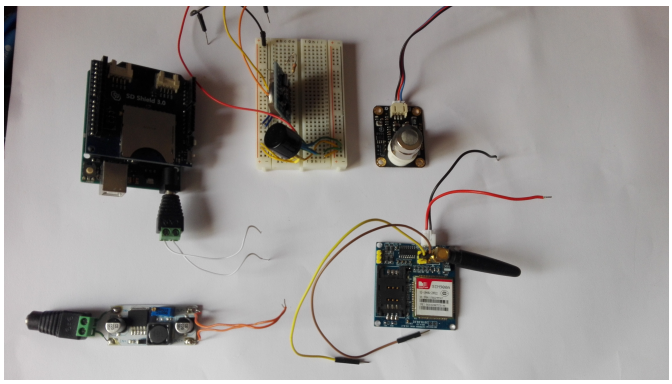


Figura 3.2.1: Componentes adquiridos

A bateria não se encontra na figura mas fornece 12 V e 2 A que não são problema para o LM2596S que consegue converter qualquer tensão e corrente de entrada de 3 V a 40 A e um máximo de 2.5 A em qualquer tensão de saída de 3 V a 40 A. Já os cabos não funcionam da mesma forma portanto convém verificar se as dimensões dos cabos fazem sentido para a corrente elétrica que por eles passa. Pensando na clássica analogia da água aplicada a circuitos, se tivermos muita corrente elétrica a passar por um cabo de pequenas dimensões, a resistência vai ser demasiado grande, o que irá causará grande aquecimento e possível quebra do cabo[30].

Tabela 3.2.2: Consumo dos componentes

Componente	Deteção	Amplitude	Consumo (mA)
DHT11	Temperatura e Humidade Relativa	0-50^{o}C e 20-80% HR	2.5
MG-811	CO_2	350-10000 ppm	200
MQ-131	O_3	10-1000 ppb	200
GSM/GPRS SIM900A	N/A	N/A	590
SD Shield 3.0	N/A	N/A	100
Ventoinha	N/A	4.5-13.8 V	110
		TOTAL	1203

3.3 Comunicação Hardware-Software

Após decidido que componentes fazem parte do protótipo a construir, é fundamental fazer a comunicação entre esses componentes e o Arduíno. Na figura 3.3.1 podemos ver o esquema final do protótipo que passou por diversas iterações até chegar à sua forma final presente e, nesse mesmo esquema, podemos ver que o SD Shield 3.0 se encontra montado, servindo apenas como um backup de dados no caso da comunicação GSM/GPRS falhar. Ainda nesta secção será apresentada a teoria que permite chegar ao esquema final.

Tomando a figura do Fritzing como referência e para assegurar o bom funcionamento de todos os componentes com o Arduíno, é preciso que a cada código utilizado ou criado seja atribuído um terminal de comunicação I/O, de modo a que cada componente tenha o seu lugar na comunicação com o Arduíno sem que os sensores se atropelem uns aos outros.

3.3.1 Hardware

Para se fazer a comunicação física entre o Arduíno e os componentes é fulcral conhecer o *layout* dos terminais do Arduíno e os seus protocolos de comunicação. Na figura 3.3.2 podemos ver representado o layout do Arduíno e seguidamente será explicado a que terminal corresponde cada protocolo de comunicação. Toda a informação referente a esta secção pode ser encontrada na página oficial do Arduíno entre outras[31].

Neste layout, podemos ver alguns acrónimos que fazem parte dos protocolos de comunicação utilizados entre o Arduíno e o componente para o qual se está a tentar enviar ou receber informação.

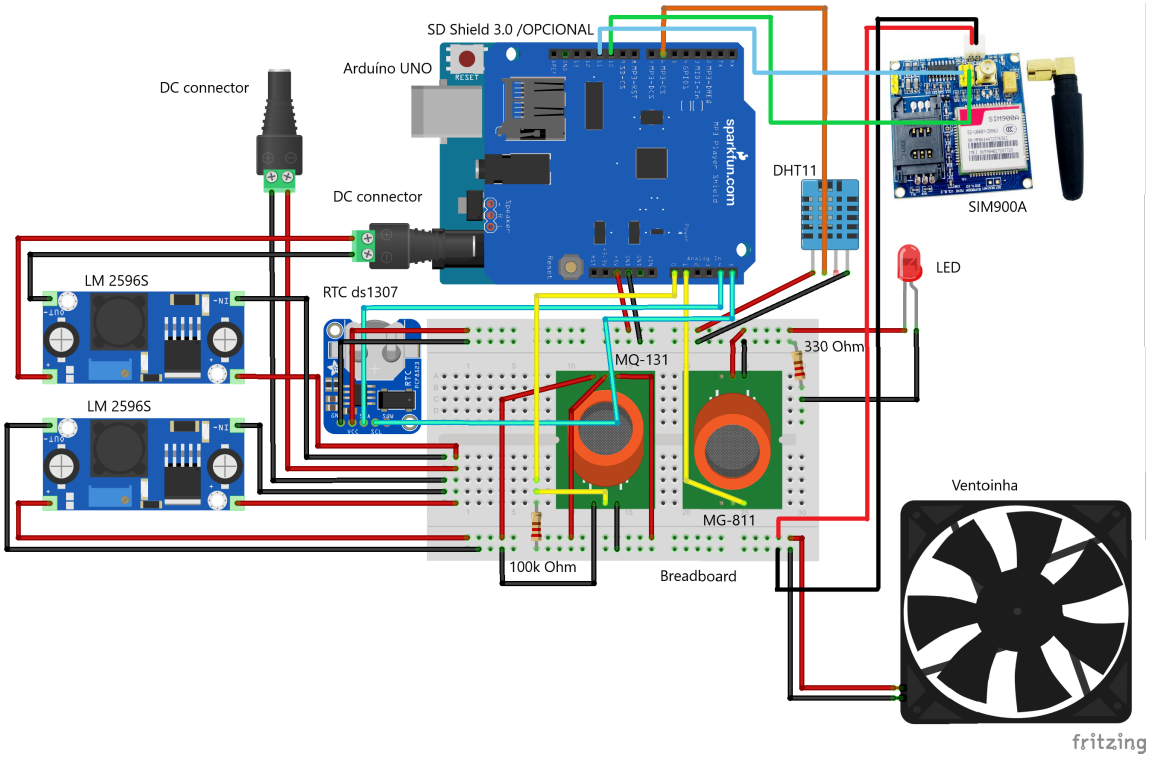


Figura 3.3.1: Esquema final da ligação completa

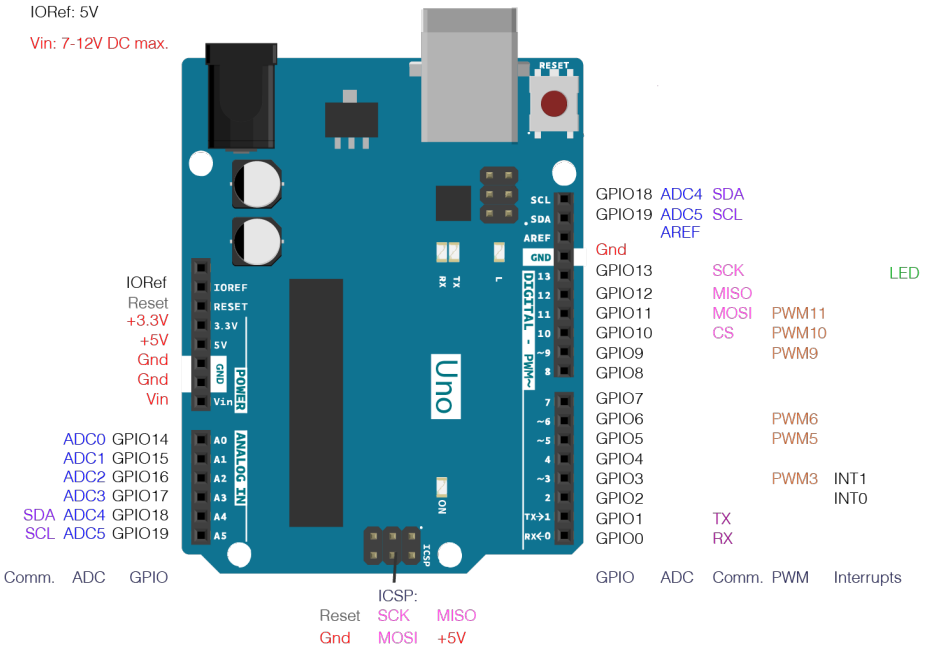


Figura 3.3.2: Layout dos terminais do Arduíno

Estes protocolos funcionam através de dois tipos de comunicação, síncrono e assíncrono. Mais especificamente, cada sensor utilizou um determinado pino GPIO (General Purpose Input/Output).

Os terminais I/O A0 e A1, foram utilizados para a comunicação analógica, sendo esta comunicação mais lenta que a digital mas é útil para se registar valores de tensão, como por exemplo, os valores de tensão do MG-811 e MQ-131. Quando um valor de tensão de saída do sensor é registado, toda essa informação é emitida ao Arduíno através de um sinal síncrono que é traduzido num valor entre 0 e 1023 bits.

Os terminais I/O A4 e A5, SDA(*Serial Data Line*) e SCL(*Serial Clock Line*), também síncronos, são denominados de comunicação I2C (*Inter-Integrated Circuit*) e são um tipo de comunicação de dois sentidos, que possibilita enviar informação do computador para o componente, neste caso o RTC. A informação enviada consiste na data e hora no momento do registo, feito pelo RTC, que guarda e procede ao seu registo contínuo operando como um relógio. Posteriormente é utilizada essa contagem contínua para se efetuar um timestamp, ou seja, cada vez que os sensores enviam a sua informação para o Arduíno, essa informação é enviada para o SD card com a Data e hora referente a esse registo.

Os terminais I/O de 13 a 10 são terminais de comunicação SPI (Serial Peripheral Interface). São também um tipo de comunicação síncrona, que através de um relógio de sinal oscilante, diz ao recetor exatamente quando e como deve orientar os bits da amostra que quer registar. Isto permite também comunicação em dois sentidos, assim como vários dispositivos ligados ao mesmo *Master*, que estabelece o sinal *Clock* que os dispositivos *Slave* devem seguir. Com o Arduíno a servir de Master, é possível ter um ou mais Slaves mas, neste caso, temos apenas um, o SD shield 3.0. Este SD shield 3.0 permite que o Arduíno envie a informação referente aos sensores para o mesmo que os guardará no SD card.

No pino digital I/O 6 temos o DHT11, que nos é aconselhado pelo datasheet do sensor a utilização de um pino digital. Este sensor já está inserido num módulo que possibilita a sua utilização com alimentação de 5 V mas, caso não estivesse, seria necessário fazer um pull-down da tensão com uma resistência de 5 kΩ, é comum este sensor vir sem módulo pelo que se aconselha a compra com módulo para reduzir a complexidade do circuito.

Seguidamente temos a comunicação UART (*Universal Asynchronous Receiver-Transmitter*). É uma comunicação assíncrona que recolhe bytes de dados e transmite os seus bits individualmente de forma sequencial, semelhante ao SPI, mas com um funcionamento diferente. No caso do SPI, em que tanto o transmissor como o recetor partilham o mesmo *clock timing* para se entenderem, o UART pode ter diversos timings diferentes para diversos bytes de dados diferentes, sendo que contém depois um microcontrolador secundário que reconstrói a informação para o devido timing a ser interpretado pelo Arduíno. Com a inclusão da *SoftwareSerial Library* é possível utilizar a grande maioria dos terminais digitais I/O como meios de comunicação UART, onde neste caso utilizou-se o pin 5 e 2 para a comunicação do GSM.

3.3.2 Software

Através do Arduíno IDE (*Integrated Development Environment*) são carregados para o mesmo os códigos utilizados e criados de modo a estabelecer as instruções que este deverá seguir. Esta parte apresentou alguns desafios e recomenda-se um bom conhecimento base de C ou C++ para se poder fazer estas compilações de código. Juntamente com o Arduíno IDE foi utilizado o Notepad++ ou o CodeBlocks para auxiliar nas leituras e alterações de códigos recolhidos nos repositórios do Github.

Considera-se que existem dois códigos finais, ambos presentes no Anexo A.1, um deles sem a aplicação de GSM/GPRS e a enviar dados para o cartão SD através do SD Shield 3.0, e o outro com aplicação do GSM/GPRS a enviar dados para a *cloud*. Facilmente se combinam os dois embora não tenha sido tentado, mas não implica mais do que compilar os dois códigos em conjunto, testar e despistar possíveis problemas.

Com o código sem GSM/GPRS, ou seja, a primeira versão do código dito final, é possível configurar o Arduíno para enviar a informação referente aos sensores MG-811, MQ-131, e DHT11 diretamente para um ficheiro *Excel* no cartão SD, com um timestamp emitido pelo RTC, que é agregado aos valores recebidos dos sensores correspondentes a esse segundo. Mais tarde, ao trabalhar os dados no Excel, foi possível observar algumas incongruências dado que os valores de vez em quando saltavam um segundo. A sua explicação reside numa nuance causada pelo tamanho do código, assim como inerentes complexidades associadas à utilização de *Strings* em vez de *Chars*, que complicam um pouco o processamento da informação. Como o código ocupa memória no Arduíno, pois está-se a tentar usar diversos sensores ao mesmo tempo com um só código, a SRAM demora uns milissegundos a mais a processar a informação, o que causa com que o RTC não consiga por vezes fazer o timestamp de 1 em 1 segundo como requerido pelo código. É possível com vontade e conhecimento, podar o código para circundar estas limitações.

3.4 Construção do prototipo

É preciso relembrar que, tanto a secção de construção do protótipo, como a de comunicação Hardware-Software, estão intrinsecamente ligadas dependendo fortemente uma da outra. Nesta dissertação, as duas subsecções encontram-se separadas como modo de tornar a leitura mais estruturada, porém, em termos de trabalho prático, ambas foram progredindo em conjunto.

A fase de construção passou por várias modificações até se chegar à conclusão final, nesta secção serão representadas as situações finais de cada fase assim como reportadas as modificações mais importantes no que diz respeito ao funcionamento dos componentes.

3.4.1 Pré-Montagem do protótipo

Na fase de pré-montagem foi importante aferir o bom funcionamento dos componentes a montar no protótipo final. Desta forma, à medida que foram recetadas as encomendas com os compo-

nentes, nomeadamente os sensores, foram realizados alguns testes com estes a serem alimentados pelo Arduíno através de USB a operar a 5 V. Na maioria dos componentes, os testes são simples e envolvem simplesmente verificar se existe a passagem de corrente e se o Arduíno está a receber a sua informação através dos códigos no Anexo A.1. No que diz respeito aos sensores, não é tão simples, pois tanto no MQ-131 como no MG-811, é preciso localizar os terminais H, pois estes são os terminais que estão ligados ao filamento de aquecimento que é parte fulcral do bom funcionamento do sensor. Podemos ver onde estes terminais se localizam no MQ-131 e no MG-811, através do seu esquema partilhado retirado dos seus datasheets[32, 33]:

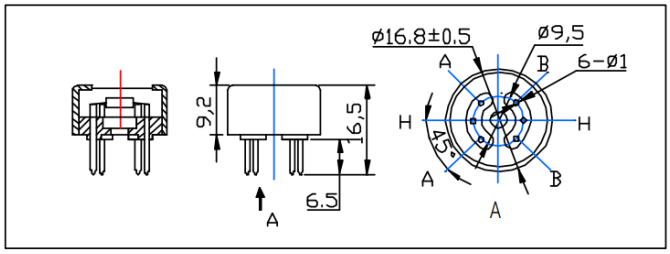


Figura 3.4.1: Desenho técnico do sensor MQ-131 e MG-811

Sabendo onde se encontram os terminais de aquecimento H, precisamos de saber a tensão esperada de operação dos mesmos. No caso do MG-811, todos os datasheets apontam para um funcionamento de 6 V, logo, é esperado que ao medir a tensão aos terminais H com um multímetro, seja medido o valor de 6 V. No caso do MQ-131 não é tão linear pois existem discrepâncias de datasheet para datasheet, assim como existe mais do que um modelo do seu género disponível, o que cria algumas dúvidas em relação à verdadeira tensão que deve ser medida. Segundo o datashet da WINSEN, o valor esperado será de 5 V, no entanto, o sensor adquirido nesta dissertação parece ser uma versão cópia do sensor da WINSEN e existem alguns datasheets que indicam funcionar a 6 V, contudo supomos que é uma cópia porque faltam alguns traços físicos do sensor original. A própria página do Arduíno faz menção aos sensores MQ não existindo certezas quanto à utilização de 5 V ou 6 V, sendo mesmo dito "sensores que usam 5 V ou 6 V..."[34], porque existem diversos sensores MQ no mercado. Com esta advertência serão seguidamente expostas as experiências efetuadas nos dois sensores na fase de pré-montagem.

3.4.1.1 Sensor MG-811

No caso do MG-811 a ligação foi simples dado que o próprio módulo vem acompanhado de um cabo molex já com as saídas VCC, GND e Analog. O módulo também tem disponível um datasheet elaborado pela empresa criadora do mesmo, a Sandbox Eletronics.

Na figura 3.4.2 temos o esquema de ligação do módulo do sensor MG-811 visto de baixo, assim como a localização dos terminais H onde temos de colocar as pontas de prova do multímetro, com o intuito de despistar qualquer problema de aquecimento.

De facto, com o USB do PC a alimentar o Arduíno que por sua vez alimenta o módulo com 5 V, são registados os 6 V esperados nos terminais H, o que indica que o módulo está a funcionar de

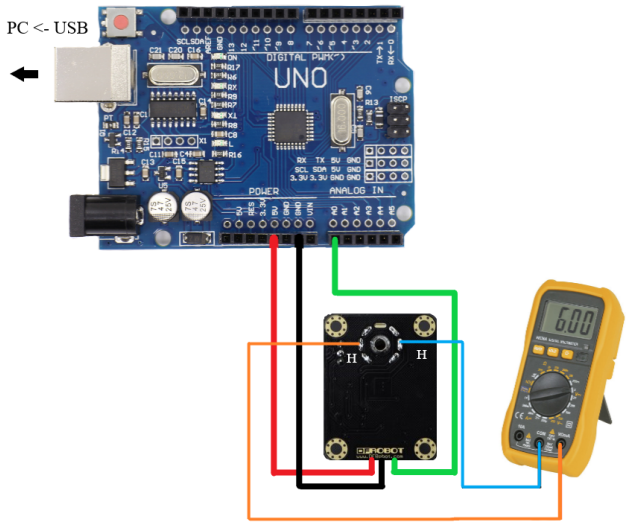


Figura 3.4.2: Ligação teste do sensor MG-811

forma expectável. Seguidamente, com o código simples do Arduíno no Anexo A.1, podemos verificar que as tensões de saída registadas no serial monitor do Arduíno IDE também fazem sentido. Dizemos que fazem sentido porque registam valores semelhantes às tensões de fábrica. É importante fazer esta comparação, porque podemos observar na figura 3.4.3 que precisamos de cerca de 0.8 V para cobrir toda a amplitude de medição do sensor. Se a nossa tensão de saída medida fosse, por exemplo, de 0.8 V, significaria que seria possível chegar aos 0 V sem estarmos a cobrir toda a amplitude do sensor.

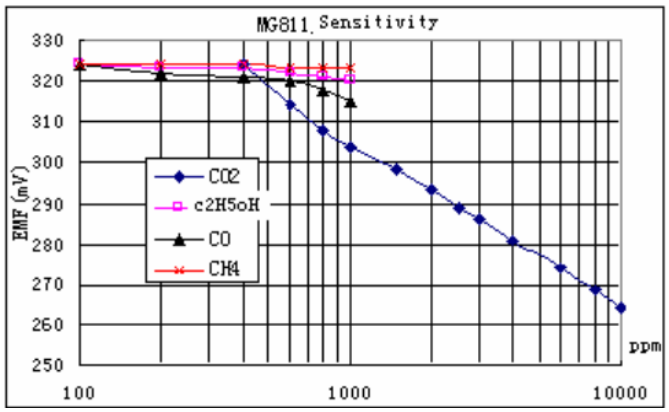


Figura 3.4.3: Amplitude de medição do sensor MG-811

3.4.1.2 Sensor MQ-131

Os testes ao MQ-131, também montado num módulo, foram iniciados com alimentação de 5 V à semelhança do MG-811. Seguidamente mediu-se com um multímetro a tensão aos terminais H. A tensão medida pelo multímetro registou 5 V, o que aparenta justificar que o datasheet correto

será o dos 5 V e não o dos 5 V, no entanto, quando se fez o teste para aferir a tensão de saída do sensor registada pelo Arduíno, através do código no Anexo A.1, foi evidente que os valores de tensão de saída com alimentação a 5 V estavam demasiado baixos, próximos de 1 V, o que pode revelar-se um problema visto que estes testes de tensão foram feitos em dia nublado e em ambiente interior fechado, ou seja, o ozono presente no ar estaria perto de 0 e assim que se começasse a fazer medições de ozono, os valores de tensão iriam descer e seria possível chegar aos 0 V sem estarmos sequer próximos do verdadeiro limite inferior da amplitude de medição do sensor. Em conjunção com este facto, ao tentar manipular-se a tensão de saída com o potenciómetro do módulo, inferiu-se que este aparentava não ter qualquer efeito. Daqui retiraram-se duas ilações: ou o módulo não estava a funcionar como era devido, ou a tensão de alimentação não era a correta. Nesta fase, para à partida se despistar futuros problemas, decidiu-se que se ia abandonar completamente o módulo, utilizando o sensor só por si.

Como referido na secção dos critérios de pesquisa, muitas vezes o facto do sensor vir num módulo pode ser problemático, porque pode não existir um datasheet a explicar de que modo o módulo atua no sensor, porém, quando retiramos o sensor do módulo, a situação por norma fica mais simples. Com o sensor só por si e, recorrendo à secção do website do Arduíno que faz referência aos sensores MQ[34], assim como ao datasheet do MQ-131, foi encontrado e utilizado o seguinte esquema:

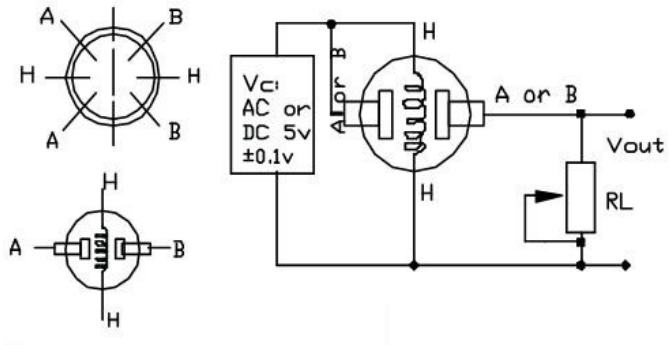


Figura 3.4.4: Ligação teste do sensor MQ-131

Segundo o datasheet do MQ-131 recomenda-se a utilização de uma resistência de carga RL (Resistance Load) de 100 k a 200 kΩ. Neste caso foi utilizada uma resistência de 100 kΩ como RL, esta deu-nos os melhores resultados nos valores de tensão de saída, cerca de 3 V, em vez dos 1 V que dificilmente cobririam a amplitude das medições.

É importante referir que no estado final desta fase de montagem, devido a alguns constrangimentos na organização da distribuição da tensão e corrente na caixa, como será discutido de seguida, optou-se por utilizar 6 V de alimentação.

3.4.1.3 Preparação da Caixa

Não poderia existir um protótipo minimamente estético sem ter um local onde colocar todos os seus componentes. Para essa finalidade foi comprada uma caixa com dimensões 159x140x60 mm.

Para servir as necessidades do projeto foi necessário alterar a caixa com a utilização de um berbequim, fazendo os buracos para as entradas de ar e uma abertura para instalar a ventoinha na parte traseira que servirá de exaustor. Foram feitos mais três buracos, um para colocar um LED de modo a indicar quando o dispositivo está ligado, outro para servir de abertura para se conectar o transformador ao DC connector do LM2596S que depois transforma os 12 V em 6 V e alimenta todo o aparato e, finalmente, outra abertura para a antena do GSM/GPRS, que foi feito mais tarde quando se ponderou a sua aplicação.

A caixa alterada e pronta para se instalarem os sensores sem a alteração do GSM/GPRS pode ser vista na figura que se segue:

(a) (b)

Figura 3.4.5: Caixa para o protótipo

No que diz respeito a alterações dos componentes, foi necessário dar uns toques finais que incluíram: Soldar terminais ao RTC para ser mais fácil liga-lo numa *breadboard*, soldar cabos de modo a tornar a gestão da cablagem mais fácil e instalar mangas nos cabos soldados de maneira a impossibilitar curtos circuitos por contacto.

3.4.1.4 Restantes componentes

Quanto ao DHT11 e ao SD Shield 3.0 não houve grandes preocupações pois os seus funcionamentos estão muito bem documentados não sendo necessário qualquer cuidado técnico em especial. Existem libraries completas com códigos exemplo disponíveis tanto no website do Arduíno como no repositório Github.

3.4.2 Montagem do protótipo

Até aqui todos os componentes, à exceção do GSM/GPRS que só foi instalado mais tarde, foram testados individualmente utilizando o USB ou o transformador como fonte de alimentação. Foi ainda feito o testo aos sensores DHT11, MQ-131 e MG-811 ao mesmo tempo, funcionando

sem problemas, porém, nunca foram testados todos os componentes juntos, que incluem o SD shield 3.0, o RTC e a ventoinha. Chegou altura de fazer essa montagem completa e testar o seu funcionamento durante períodos de tempo prolongados.

Nesta fase do projeto onde todos os componentes estavam ligados ao mesmo tempo, foi testada a alimentação através do transformador pelo DC-jack do Arduíno a 6 V.

Com todos os sensores ligados ao mesmo tempo começaram a surgir sintomas do mau funcionamento do MG-811. A luz do potenciómetro do sensor acendia, alertando para um problema de alimentação. Quando tal acontecia, verificava-se a tensão aplicada ao terminais H e denotava-se que a tensão registada não eram os esperados 6 V, mas sim 4 V, o que justificava que algo estava realmente errado na alimentação. O Arduíno não estava a aguentar com a carga e não estava a conseguir fornecer os 5 V ao MG-811, que são depois amplificados pelo módulo do sensor em 6 V para alimentar os terminais H. Inicialmente não foi encontrada a causa do problema e foram testadas algumas hipóteses como religar o sensor e até modificar o seu posicionamento dentro da caixa o que, curiosamente, funcionou algumas vezes fazendo a tensão aos terminais H retornar ao normal. Porém, esta situação ocorreu demasiadas vezes, arruinando tentativas preliminares de calibração.

Até aqui ainda se desconheciam os limites de corrente elétrica e tensão que se deve fornecer ao Arduíno.

Partindo dessa ignorância foi decidido testar-se algumas hipóteses para despistar o problema de alimentação. Optou-se pela mudança da fonte de alimentação do sensor, que até agora vinha do Arduíno, para passar a ser alimentado diretamente pelo transformador. Desta forma, foi novamente reverificada a tensão aos terminais de aquecimento H com o multímetro e mostravam 6 V, como esperado, pois ao contrário do Arduíno em sobrecarga, a tensão e corrente vinda diretamente do transformador era constante, se bem que, estávamos a fornecer diretamente 6 V ao módulo em vez dos 5 V recomendados. Foi deixada assim a configuração durante esta fase e o problema foi resolvido, a luz do potenciómetro não retornou a acender nas restantes experiências.

Foi uma situação de erro que não deveria ter acontecido pois este lapso deveu-se à falta de conhecimento dos limites do Arduíno e o ideal teria sido comprar mais um conversor DC-DC LM2596S de modo a transformar os 12 V da bateria em 7.5 V e alimentar o Arduíno. Com o segundo LM2596S transformaríamos os 12V em 5 V para fornecer os sensores, deste modo o Arduíno não teria tido problemas em alimentar o MG-811. Infelizmente as calibrações foram realizadas com os sensores a serem alimentado por 6 V, o que pode ter tido alguns impactos que serão discutidos nos resultados e conclusões.

Esta situação de erro deu origem ao esquema presente na figura 3.4.6. Esta configuração esquemática, apesar de completamente errada de um ponto de vista eletrónico devido a um mau planeamento da distribuição de tensão, tem em conta outros fatores não relacionados com a parte elétrica mas que se consideraram certos. Essas considerações foram reutilizadas na fase final de montagem do protótipo com o GSM/GPRS.

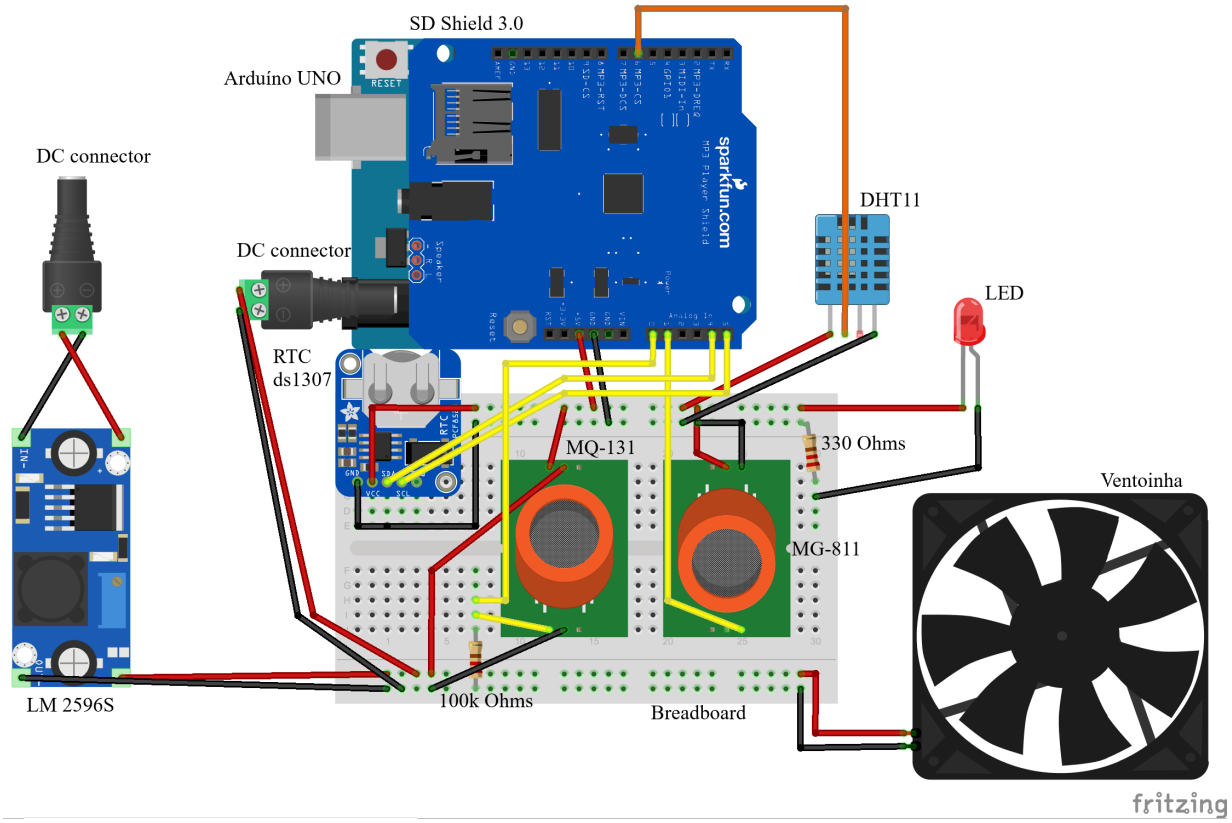


Figura 3.4.6: Primeiro esquema de ligações do protótipo

Essas considerações referem-se à grande quantidade de calor produzida pelos sensores MQ-131 e MG-811, onde a solução passou por posicionar os sensores relativamente próximos da ventoinha de extração do ar, o que ajuda a dissipar o calor dentro da caixa. Outro fator que não pode ser observado na figura é o facto de os sensores serem colocados com uma inclinação de 45 graus. O topo do sensor deve estar a apontar na direção oposta ao fluxo de ar para, que deste modo, seja possível o ar fluir livremente por dentro do sensor. O sensor recebe a amostra de ar na sua parte superior e esse ar flui saindo pela parte inferior. Tem como vantagem não possibilitar a estagnação de uma amostra de ar dentro do sensor ou acumular humidade.

3.4.3 Protótipo final em comunicação com cloud

Para finalizar e dar a conhecer aquela que foi efetivamente a configuração final depois de todos os problemas na construção, foi o momento de implementar o GSM/GPRS. A fase de implementação do GSM/GPRS ocorreu depois dos testes de calibração e juntamente com esta montagem foi também corrigido o erro prévio de alimentação do Arduíno, situação que será discutida nas conclusões.

Já havia sido ponderada a implementação de comunicação 2G, 3G ou 4G de modo a tornar o protótipo completamente wireless. À medida que os trabalhos foram progredindo a ideia foi deixada para trás, no entanto, a metodologia que estava presente de ter um cartão SD dentro da caixa, faz

com que sempre que seja preciso ir buscar dados para analisar e verificar as medições das calibrações, é necessário abrir o protótipo, o que se revelou ser muito pouco prático. Foi colocado em movimento o estudo de implementação de GSM/GPRS de modo a poder fazer upload aos dados em tempo real para uma cloud network, em português, uma rede de computação em nuvem. O dispositivo escolhido, como enunciado previamente foi o módulo SIM900A, módulo esse que é extremamente barato e fácil de obter. O GSM/GPRS SIM900A é um módulo 2G que opera na dupla banda de frequências 900 e 1800 MHz.

Nos Estados Unidos já não é possível utilizar a rede 2G pois os operadores decidiram que precisam dessas frequências para a rede 4G, no entanto, na Europa essa mudança não é tão clara e está a acontecer de forma mais lenta em apenas alguns países. Em Portugal ainda existe uma boa cobertura de rede 2G. Fez-se este pequeno parênteses sobre as redes, porque mesmo que a rede 2G desaparecesse em Portugal, seria facilmente substituído o módulo 2G por um 3G ou 4G dado que estes módulos operam da mesma forma através de comandos AT(Hayes Modem Commands) e são normalmente da mesma empresa, a SIMCOM.

Infelizmente, no que diz respeito a componentes eletrónicos de baixo custo, apesar de ter sido feito uma pesquisa preliminar, nem tudo é linear e só após a receção do SIM900A é que surgiram algumas complicações. Seguidamente irá ser feita uma decomposição de todo o processo de montagem e funcionamento do GSM/GPRS.

3.4.3.1 Configuração e testes com o GSM/GPRS

Foi iniciada a ligação entre o Arduíno e o GSM/GPRS, mas rapidamente se chegou à conclusão, que sem entender o funcionamento dos comandos AT, seria complicado criar o código para automatizar a comunicação. Desta forma, para nos familiarizarmos com o GSM/GPRS, optou-se por primeiro fazer uma comunicação simplificada do componente e o computador via UART, através de um mini-USB TTL. O esquema da ligação é o seguinte:

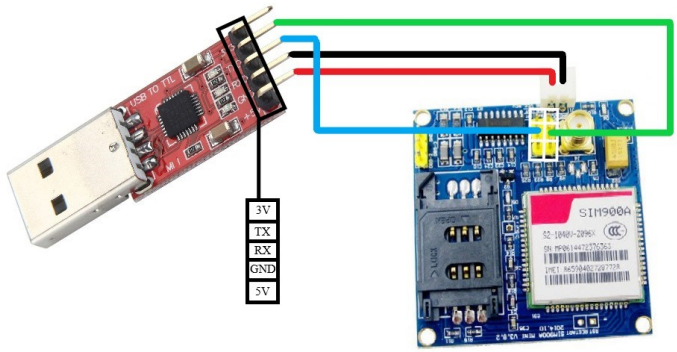


Figura 3.4.7: Ligação mini-USB TTL e módulo SIM900A

Mesmo sem o cartão SIM inserido, é possível testar o módulo GSM/GPRS e os seus comandos AT sem ter de lidar diretamente com a programação em C no Arduíno.

Para saber se o módulo está a funcionar devidamente, testamos os comandos AT no programa Terminal v1.9b. O primeiro comando a enviar é o comando "AT" (Attention), que nos garante se o componente está a funcionar de forma devida. Este deverá responder-nos "OK" na janela do terminal como podemos observar na figura que se encontra abaixo:

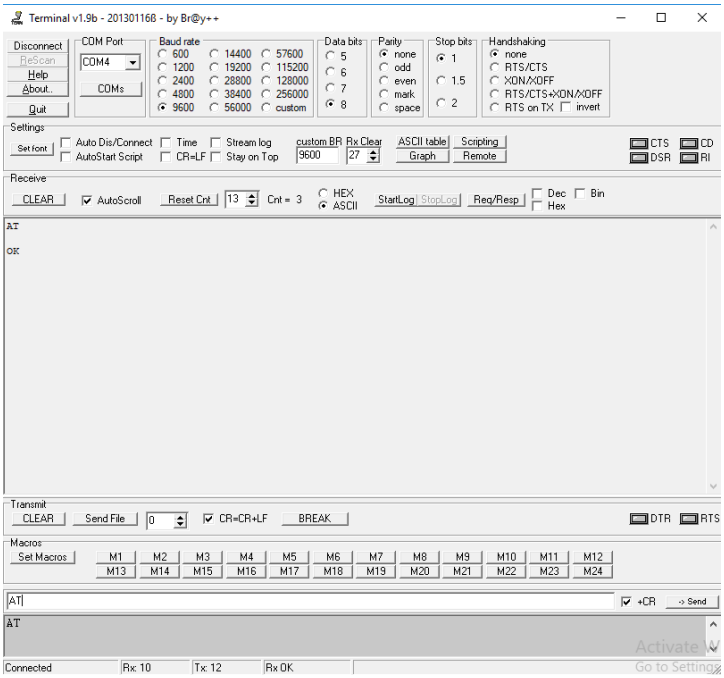


Figura 3.4.8: Teste de comunicação com o módulo SIM900A

Num segundo teste, com o cartão SIM inserido desta vez, surge a primeira complicação do SIM900A. Embora se tenha obtido a resposta "OK" ao comando de teste AT, que implica que o módulo está a funcionar devidamente e efetivamente a comunicar com o computador, qualquer outro comando, como por exemplo AT+CMGS=1, que dá início à comunicação SMS, dá-nos o resultado ERROR. Após a leitura de alguns *blogs* foi encontrado um em específico[35] que indica que este modem deve apenas ser utilizado no mercado asiático, ou seja, quando se introduz um cartão SIM de uma operadora de rede móvel europeia, não é possível fazer qualquer comunicação entre o módulo e o mundo exterior. Este fator não estava especificado na página de compra nem em qualquer pesquisa preliminar. Acredita-se que isto se deva à mistura de informação entre SIM900 e SIM900A, partilhando ambos o mesmo datasheet.

A primeira coisa que foi feita para despistar o problema foi retirar o PIN do cartão SIM, para tal, a maior parte dos telemóveis de última geração, com sistema Android, tem uma opção nas suas definições chamada de *LOCK SIM CARD*, onde é possível remover por completo o PIN do cartão SIM. Após não se obter bons resultados com esta primeira abordagem, foi aprofundada a pesquisa tentando encontrar uma solução para o problema do *Region-Lock*. Foram encontradas diversas referências em fóruns Arduíno e blogs[35, 36] acerca da possibilidade de fazer *update* ao *firmware* do SIM900A com o firmware do SIM900. São recomendados, por diferentes utilizadores, diversos firmwares que podem ser descarregados no website da SIMCOM[37].

Para se proceder ao update do firmware do SIM900 no SIM900A, mantemos a conexão esquemática da figura 3.4.7, mas precisamos do programa *Costumer flash loader* disponível no website da SIMCOM.

Com o programa instalado, podemos observar na figura que se segue, diversos ficheiros que foram testados, inclusive podemos ver uma seta que aponta para o ficheiro que finalmente tornou possível a comunicação do módulo com o mundo exterior, note-se que é referente a um SIM900B de 32mb.

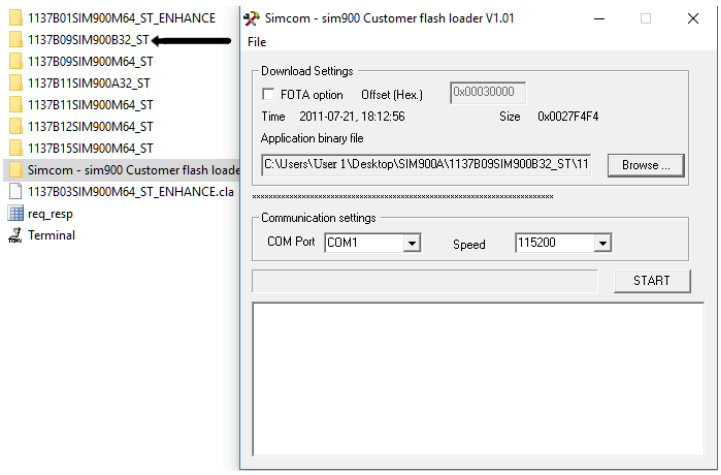


Figura 3.4.9: Update ao firmware do SIM900A

Apesar das inúmeras referências à utilização do firmware do SIM900 no SIM900A, tornou-se evidente que não estavam a funcionar, pois após a cada update de firmware, era novamente testado o módulo GSM/GPRS com o Terminal v1.9b utilizando os comandos AT, mas não se obteve a resposta OK nos testes de comunicação SMS ou Internet. Após uma nova pesquisa aprofundada dos comentários de vários utilizadores onde o update do firmware também não funcionou, tornou-se claro que o SIM900A tem uma nova versão, relativamente recente, que à diferença do chip do SIM900A de iterações passadas, não é 64mb, mas sim 32 mb[35], ou seja, qualquer firmware do SIM900, que é 64mb, não vai funcionar no chip do SIM900A. Para contornar este novo entrave foi testada a hipótese de em vez de se utilizar o firmware do SIM900, experimentar-se o do SIM900B, que à semelhança do SIM900A, também é 32mb. Com este firmware foi finalmente possível obter um "OK" à resposta do AT+CMGS=1 e proceder-se ao envio e receção de chamadas e SMS e posteriormente o envio de dados para a cloud através de um APN(*Access Point Name*).

O serviço cloud escolhido para o *proof-of-concept* foi o ThingSpeak.com, um website totalmente gratuito e bastante intuitivo de se usar. O código final que faz a comunicação entre o módulo GSM/GPRS e o Arduíno pode ser encontrado no Anexo A.1. O que este código faz resumidamente é abrir a conexão entre o módulo e o website ThingSpeak enviando informação dos sensores para o APN e fechando a comunicação de seguida. O código repete este ciclo enquanto o Arduíno e o GSM/GPRS estiverem ligados, acrescentando continuamente dados às tabelas.

Na figura 3.4.10 podemos constatar como funciona a visualização em tempo real do envio de dados. Sempre que é enviado um valor referente à tensão registada nos sensores, este é registado

num gráfico para o seu efeito no website ThingSpeak. Futuramente com os sensores devidamente calibrados, seria possível através do código no Arduíno fazer o upload diretamente dos valores em ppm.

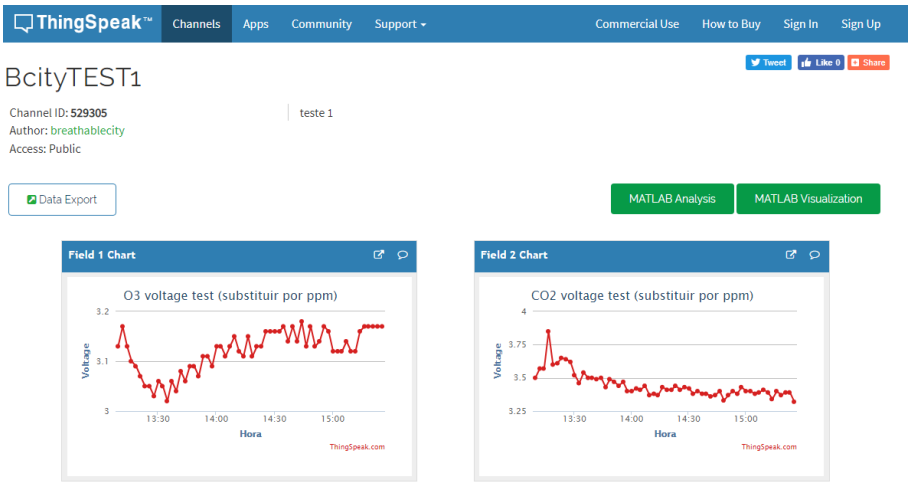


Figura 3.4.10: Valores de tensão enviados através do GSM/GPRS para o ThingSpeak

3.4.4 Análise de Custos

Reconhece-se que existem alguns artigos que exploram bem o conceito de low cost, até investindo em placas PCB ou circuitos integrados, reduzindo não só os custos mas também as dimensões do protótipo, no entanto, dado que este estudo teve como motivação a iniciativa Breathable Cities, vai ser utilizado o exemplo mais extremo, mostrando assim o potencial de redução de custos, numa comparação entre os componentes mais dispendiosos face aos mais low cost.

Tabela 3.4.1: Análise de Custos entre Breathable Cities Campaign e Protótipo

	Protótipo FCUL			Breathable Cities	
Sensor	Deteção	Preço (€)	Sensor	Deteção	Preço (€)
DC1700	Partículas ultrafinas	408	MG-811	CO_2	45.9
EL-USB-CO	CO	111	DHT11	Temperatura e HR	3.7
Dusttrak 8533	Partículas	9,655	MQ-131	O_3	48.8
Aethlabs AE51	Black Carbon	9,655	.....	.....	....
	TOTAL	19,829		**TOTAL**	98.4

Podemos rapidamente depreender que os sensores não detetam os mesmos poluentes, de qualquer das formas, é interessante demonstrar um caso extremo versus um caso low cost. Em futuras dissertações é possível facilmente incluir um sensor de partículas que tem um custo de 25 euros, que quando comparado com o AE51 de 9655 euros, se torna numa diferença notável.

Como mencionado acima, no caso mais extremo de uma estação de monitorização, podemos observar uma redução de cerca de 99,5% do custo.

Capítulo 4

Calibrações

A calibração é a comparação de um equipamento ou padrão de medição de precisão desconhecida versus um equipamento de referência. No caso desta dissertação, foram comparados os dados das medições efetuadas pelos sensores MG-811 e MQ-131, com os instrumentos de referência Telaire e Aeroqual, com intuito de detetar, correlacionar, documentar e eliminar, através de ajustamento, qualquer variação de precisão.

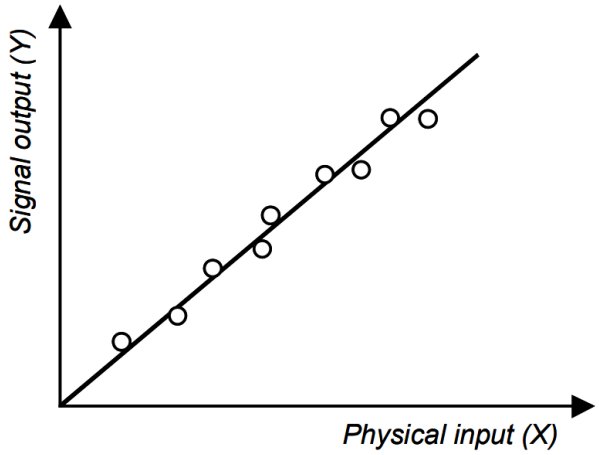


Figura 4.0.1: Curva típica de calibração

Com a primeira versão do protótipo montada, seguindo o esquema da figura 3.4.6, sem o GSM/GPRS montado, este estava pronto a ser testado na integra com todos os sensores no seu interior.

4.1 Metodologias de calibração

Desde o inicio que se definiu que iam ser realizadas medições de 2 a 5 dias e consideradas como um *set* de dados. Esta decisão teve como propósito simular as medições durante um período semanal de trabalho normal. Estes dados recolhidos foram inicialmente analisados em Excel de modo a despistar qualquer mau funcionamento dos sensores. Posteriormente foram trabalhados através de análise gráfica em *MATLAB*.

As medições decorreram durante o dia e noite em períodos individuais de 4 horas. Foram efetuadas cerca de 3 a 4 medições por dia que totalizaram 12 a 16 horas. Por norma as medições vão ser executadas em períodos onde a concentração de CO_2 sobe ou desce, ou seja, para se simular uma subida de concentração de CO_2 serão feitas calibrações durante o período da noite, com um ocupante no quarto com tudo fechado e, deste modo, foi possível fazer medições dos 400 aos 2000 ppm com alguma facilidade. Para simular um período de aumento de concentração foram realizadas calibrações durante a manhã seguinte, com o quarto desocupado e a janela ligeiramente aberta, de modo a que se acompanhasse a dissipação gradual do CO_2.

Todas as medições produzidas pelo MG-811, MQ-131, Telaire e Aeroqual 500 foram gravadas em ficheiros Excel e posteriormente trabalhadas em MATLAB. Os dados recolhidos do MG-811 e MQ-131 foram guardados através do SD Shield 3.0 no cartão SD. Os dados do Telaire foram gravados através de um *DataLogger* e o Aeroqual tem o seu próprio DataLogger interno que tornou possível exportar os seus dados para um ficheiro Excel. É importante referir que estes ficheiros têm de ser CSV (Comma-Separated Values), para posteriormente serem utilizados no MATLAB.

Outro fator importante a ter em conta é o tempo de resposta dos sensores. O Telaire, segundo o seu datasheet[38], tem 10 segundos de tempo de resposta, ou seja, demora 10 segundos a detetar uma mudança de CO_2 no ar ambiente. Quanto ao MG-811, com a pouca informação encontrada acerca do seu tempo de resposta, considerou-se instantâneo. Podemos observar em baixo a única figura encontrada a seu respeito:

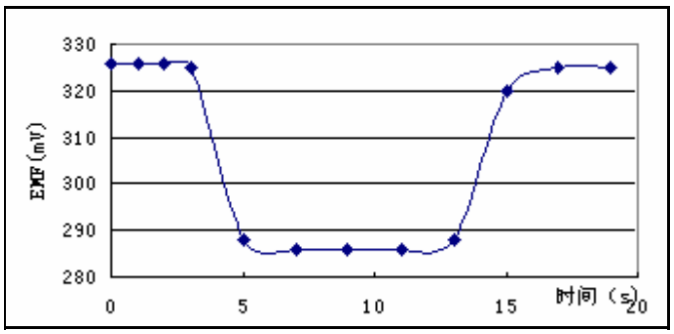


Figura 4.1.1: Tempo de resposta do sensor MG-811

Esta figura representa o sensor a ser exposto momentaneamente a um ambiente onde o gás a medir está presente. É detetada uma mudança que demora cerca de 1 segundo a processar completamente e, finalmente, demora cerca de 8 segundos a resumir, ou seja, demora 8 segundos a voltar à tensão inicial quando o gás não estava presente.

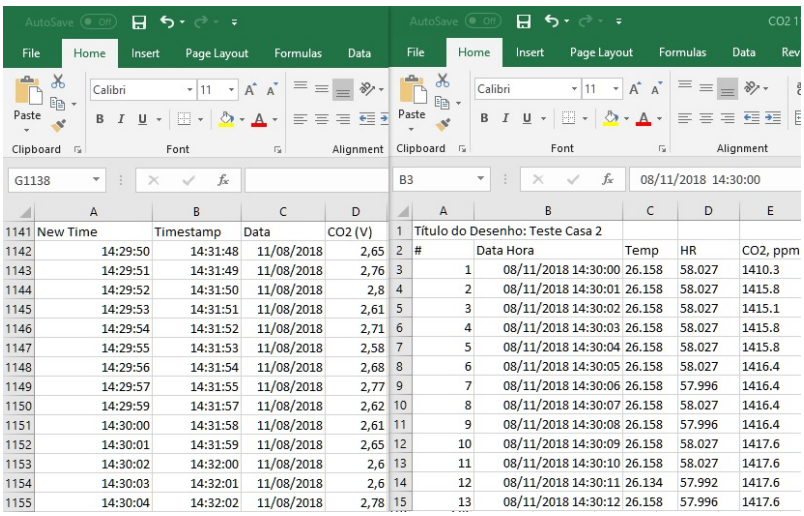
No caso do Aeroqual 500 a situação é semelhante, mas em vez de 10 segundos este diz-nos, no seu datasheet, que um T90 de 40 segundos[39]. O tempo de resposta T90 é o tempo necessário para o sensor detetar 90% do nível máximo da concentração do gás no ar, ou seja, se por exemplo tivéssemos um barril com água quente, ao abrirmos uma torneira que se encontra á temperatura ambiente, o tempo T90 seria o tempo que demoraria para a torneira ficar a 90% da temperatura da água quente que vem do barril. No entanto, este valor foi considerado redundante pois o sensor só

consegue registar valores de 1 em 1 minuto, dessa forma, considerou-se que quando o Aeroqual 500 nos fornece um valor, este equivale ao minuto anterior.

Por esta razão, todos os dados recolhidos em Excel cuja referência é o Telaire ou o Aeroqual, que posteriormente foram trabalhados como vetores em MATLAB, exigiram que fosse realizado um ajuste de 10 segundos ou 1 minuto de modo a que as medições efetivamente correspondessem às do MG-811 e MQ-131, que foram considerados como instantâneos.

Devido aos problemas do RTC ds1307 referidos na subsecção do RTC, sempre que se acaba um set de medições e se retira o cartão SD, é necessário apontar a hora, minuto e o segundo exato em que é removido, para que se possa depois no Excel acertar as horas.

Para melhor demonstrar o contexto dos ajustes dos tempos e dos problemas do RTC, analisemos a seguinte figura:



	A	B	C	D
1141	New Time	Timestamp	Data	CO2 (V)
1142	14:29:50	14:31:48	11/08/2018	2,65
1143	14:29:51	14:31:49	11/08/2018	2,76
1144	14:29:52	14:31:50	11/08/2018	2,8
1145	14:29:53	14:31:51	11/08/2018	2,61
1146	14:29:54	14:31:52	11/08/2018	2,71
1147	14:29:55	14:31:53	11/08/2018	2,58
1148	14:29:56	14:31:54	11/08/2018	2,68
1149	14:29:57	14:31:55	11/08/2018	2,77
1150	14:29:59	14:31:57	11/08/2018	2,62
1151	14:30:00	14:31:58	11/08/2018	2,61
1152	14:30:01	14:31:59	11/08/2018	2,65
1153	14:30:02	14:32:00	11/08/2018	2,6
1154	14:30:03	14:32:01	11/08/2018	2,6
1155	14:30:04	14:32:02	11/08/2018	2,78

	A	B	C	D	E
1	Título do Desenho: Teste Casa 2				
2	#	Data Hora	Temp	HR	CO2, ppm
3	1	08/11/2018 14:30:00	26.158	58.027	1410.3
4	2	08/11/2018 14:30:01	26.158	58.027	1415.8
5	3	08/11/2018 14:30:02	26.158	58.027	1415.1
6	4	08/11/2018 14:30:03	26.158	58.027	1415.8
7	5	08/11/2018 14:30:04	26.158	58.027	1415.8
8	6	08/11/2018 14:30:05	26.158	58.027	1416.4
9	7	08/11/2018 14:30:06	26.158	57.996	1416.4
10	8	08/11/2018 14:30:07	26.158	58.027	1416.4
11	9	08/11/2018 14:30:08	26.158	57.996	1416.4
12	10	08/11/2018 14:30:09	26.158	58.027	1417.6
13	11	08/11/2018 14:30:10	26.158	58.027	1417.6
14	12	08/11/2018 14:30:11	26.134	57.992	1417.6
15	13	08/11/2018 14:30:12	26.158	57.996	1417.6

Figura 4.1.2: Manuseamento dos dados em Excel

Na figura observamos uma conjunção de duas folhas Excel. Na folha de cálculo da esquerda, que pertence às medições recolhidas pelo protótipo, podemos constatar que foi necessário retirar 118 segundos ao RTC através da subtração de TIME(0;0;118) à célula original do Tempo. Todo este tempo retirado não se deve ao RTC perder a noção do tempo mas sim, na sua maior parte, ao desajuste dos dados que dependem do código do Arduíno e do impacto da sua complexidade na SRAM, que causa a falha do registo em 1 segundo, como podemos verificar na célula 1149-1150. Como estamos a falar de milhares de dados, eventualmente perdemos 118 segundos.
Para o tratamento dos dados no MATLAB o fator da perda de segundos não tem importância pois, Como temos o tempo inicial e final das medições dos dois sensores exatamente ao mesmo tempo, a única coisa que temos de fazer é o ajuste e posteriormente interpolar o vetor com mais dados para o tamanho do vetor com menos dados.

O código utilizado em MATLAB para fazer essa interpolação e consequente modelação dos dados encontra-se no Anexo A.1. Para se utilizar as capacidades gráficas do código mencionado, foi necessário recorrer à ferramenta export_fig que pode ser consultado na página do *MathWorks*[40].

É fundamental seguir as instruções de instalação e fazer download ao *ghostscript*. Posteriormente poderemos aplicar métodos gráficos mais avançados seguindo alguns exemplos colocados pelo autor no repositório github.

Novamente referente à experiência de calibração propriamente dita, seria interessante fazer testes como o da repetibilidade, mas infelizmente não se considerou existirem condições que tornassem possível recrear as experiências de forma 100% igual umas às outras como um teste deste género requer, pois o local usado para o teste dos sensores não tem controlo da temperatura ou humidade relativa, sendo que cada experiência está à mercê da temperatura ambiente. Também não foi possível controlar as quantidades de gases presentes no ar, estando novamente à mercê do ar ambiente e das quantidades dos gases a medir nele presentes, basicamente, foi uma calibração efetuada em ambiente real cujo única maneira de combater a variância foi realizar o máximo número de medições possíveis.

As ultimas considerações das calibrações referem-se aos próprios gases a medir. Sabemos pela lei de Avogadro que a densidade do CO_2 e o O_3 no ar é de cerca de $44.01\,g\,mol^{-1}$ e $48\,g\,mol^{-1}$ respetivamente. No entanto o próprio ar atmosférico, constituído por nitrogénio, oxigénio, árgon e outros, tem uma densidade de $28.97\,g\,mol^{-1}$, logo, com estes factos, é possível denotar que tanto o CO_2 como o O_3 são cerca de 1.5 vezes mais densos do que o ar. Apesar de tudo, num ambiente comum, seja num escritório ou na rua, existem diversas variáveis, como por exemplo gradientes de pressão ou temperatura que não permitem que o gás afunde livremente em relação ao ar ambiente. As medições tiveram estas ponderações em conta de modo a posicionar os sensores no melhor local. O local escolhido foi a meia altura do canto de uma divisão com 2.6 m de pé, tendo sido este considerado ideal para para detetar movimentações de CO_2 e O_3 no ar ambiente sem estarem próximos de influência externas, como por exemplo perto de um aquecedor, ventoinha ou até a própria respiração humana.

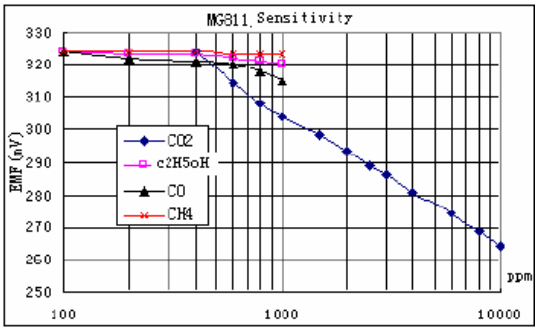
4.2 Tipos de calibrações consideradas

4.2.1 Calibração por métodos do fabricante

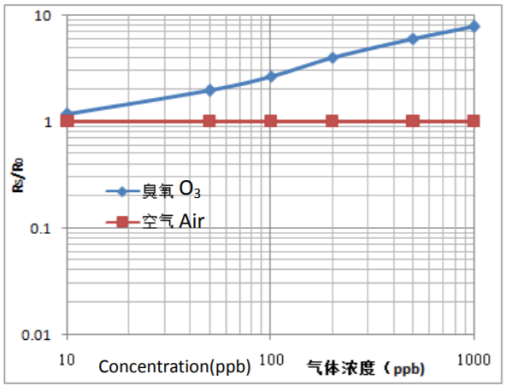
Foi considerado o processo proposto pelos fabricantes dos sensores que este passa por utilizar os sensores adquiridos tal como estão quando chegam de fábrica e aplicar as metodologias matemáticas recomendadas pelo fabricante. As equações 4.1 e 4.2.1.2 são disponibilizadas nos datasheets e podem ser encontradas as suas explicações nos métodos de calibração dos fabricantes que se seguem.

Existem grupos de entusiastas em fóruns do Arduíno e blogs que partilham as suas experiências e códigos em R ou MATLAB para se tentar calibrar estes sensores em casa. Estes códigos são úteis para quem não tiver a possibilidade ou condições de fazer uma calibração adequada. Podemos observar um exemplo desses códigos no Anexo A.1. A forma como se procede a este tipo de calibração reside na previsão do comportamento da curva logarítmica que, na melhor das hipóteses,

se comportaria da forma mais próxima possível às calibrações de fábrica dos sensores de CO_2 e O_3.



(a) Curva de calibração do MG-811



(b) Curva de calibração do MQ-131

Figura 4.2.1: Curvas de calibração de fábrica do MG-811 e MQ-131

Ficamos desde já advertidos que estas metodologias são extremamente rudimentares e que cada sensor é um sensor, ou seja, cada sensor terá sensitividades diferentes estando sempre dependente de variáveis externas e internas. Iremos seguidamente tentar entender estes processos de calibração.

Estes métodos não foram utilizados por se considerarem inferiores ao método dos mínimos quadrados porém, a única forma de aferir a sua inferioridade residiu precisamente no facto de os entender e testar algumas das suas premissas. Deste modo considerou-se importante documentá-los.

4.2.1.1 MQ-131

Para se calibrar o MQ-131 com esta metodologia de fábrica foi necessário seguir novamente o esquema sem módulo apresentado na figura 3.4.4. Como aconselhado utilizámos uma resistência RL de 100 kΩ de modo a que as tensões de saída registadas façam sentido, entre os 2 e os 3 V. Para verificarmos se os valores de tensão se encontram efetivamente entre os 2 e 3 V, ligou-se o Arduíno ao PC e foram monitorizados os registos de tensão output analógica através do Serial Monitor do Arduíno IDE. Relembramos que se utilizou o esquema de distribuição errado como podemos ver na figura 3.4.6.

Para o próximo passo e assumindo que se está a fazer este tipo de calibração porque não existe outra alternativa, seria necessário recorrer a um sensor de O_3 público, como por exemplo o da aqicn de Lisboa[41] e iremos também assumir que o nível médio de O_3 medido nesse sensor é o nível que estamos a presenciar no local onde estamos a testar o MQ-131. No caso desta dissertação nesta parte foi utilizado o Aeroqual como referência.

Em tempo real, assim que se leu o registo da quantidade de O_3 em ar ambiente, numa estação de monitorização publica, ou no Aeroqual, foi feita uma medição com o multímetro exatamente ao

mesmo tempo do valor registado aos terminais H do sensor MQ-131, valor esse que representa a resistência natural do sensor a uma determinada quantidade especifica de O_3 no ar ambiente, resistência essa chamada de Rs.

O terceiro passo foi extrapolar, com a ajuda do webplotanalyzer[42], o máximo número de pontos da curva da figura 3.4.1, pois daqui retiramos pontos que modelam o comportamento logarítmico esperado do sensor MQ-131, para seguidamente ajudar a transpor o seu comportamento ao nosso próprio sensor. Isto é conseguido através de cálculos utilizando a resistência Rs que acabamos de medir.

Para finalizar inserimos estas variáveis no script R no Anexo A.1, que são: os pontos da curva do datasheet, o Rs medido a uma determinada quantidade de O_3 no ar e o mínimo e máximo ppm que o sensor consegue medir. Com estas variáveis iremos obter outras duas variáveis que representam o valor teórico de Ro (concentração a zero ppm) e os coeficientes a e b de modo a criar uma curva logarítmica de previsão do valor de ppm que se assemelha à curva de calibração de fábrica. Esta é uma equação do tipo:

$$ppm = a \times \left(\frac{R_s}{R_o}\right)^b \qquad onde \;\; b < 0 \tag{4.1}$$

A metodologia para a calibração deste sensor não foi utilizada na sua totalidade porque desde o momento em que se começou a tentar medir os valores de Rs notaram-se grandes flutuações no valor. Várias vezes se tentou medir Rs sempre com resultados diferentes o que provocou com que o método fosse completamente abandonado. Esta metodologia é melhor descrita por David Girono no seu blog[43], pois foi ele que desenvolveu o código R utilizado para os sensores MQ.

Esta é uma calibração rudimentar que envolve muito erro experimental e que não está, mesmo assim, acessível a qualquer pessoa dado é preciso determinados conhecimentos para calibrar o sensor desta forma rudimentar.

4.2.1.2 MG-811

Para o CO_2 o método de calibração é semelhante, uma vez que a intenção é novamente gerar uma curva logarítmica, ou seja, um modelo de previsão que vá ao encontro da concentração de gás esperada consoante a tensão registada.

No caso particular do MG-811, os fabricantes dos módulos nos próprios datasheets do sensor, costumam aconselhar o mesmo método ou variações do mesmo método de calibração, onde se procura criar uma curva sabendo dois pontos dessa curva[44].

O procedimento divide-se em duas fases que estão interligadas. Na primeira fase com o Arduíno ligado ao PC e alimentado pelo transformador a 6 V, foram monitorizados os valores de tensão de saída com o objetivo de registar o primeiro de dois pontos. Ao primeiro ponto corresponde o valor médio de tensão (com duas casas decimais) quando estamos num ambiente com uma concentração

de 400 ppm de CO_2. Para atingir os 400 ppm, basta estar no exterior, ou no interior com janelas abertas e boa circulação de ar.

Seguidamente, na segunda fase, precisamos dos valores do segundo ponto. Os datasheets aconselham a medir-se os valores de tensão de saída a 1000 ppm em vez de 10000 ppm, pois é um valor mais fácil de atingir sendo que basta estar apenas um ocupante numa divisão completamente fechada e em alguns minutos é possível atingir esse valor.

Embora a equação que vamos demonstrar a seguir pareça complexa, não o é e vamos tentar explicar o seu funcionamento.

Para utilizar a equação 4.2.1.2 basta seguir atentamente o código descrito no datasheet do sensor MG-811[32]. Precisamos de introduzir no código o valor da *zero point voltage*, que é a tensão que registamos aos 400 ppm dividida por 8.5, referente ao *DC gain*. Depois precisamos de mudar também o valor da *reaction voltage* que representa a diferença entre a tensão de saída registada a 400 ppm e 1000 ppm.

$$ppm = 10^{\left(\frac{(Tensão_{registada}/DC_{Gain}) - (Tensão_{400ppm}/DC_{Gain})}{Reaction_{voltage}/(log_{10}400 - log_{10}1000)}\right) + log_{10}400} \tag{4.2}$$

Através da equação o código computa os valores de tensão de saída do sensor e transforma-os numa leitura ppm. Na equação podemos observar diversas variáveis, a tensão registada é o nosso "X", ou seja, é a tensão que está a ser registada no Serial Monitor a cada momento que o sensor está ligado a detetar variações no ar ambiente. A variável DC gain do circuito é o ganho de tensão do circuito segundo o próprio fabricante, portanto, o valor foi deixado assim e assumiu-se como uma constante. A reaction voltage a dividir pelo log de 400 e 1000 representa o declive logarítmico da nossa curva, sendo que o log de 400 no final é o nosso b. Esta é apenas uma equação da forma y=mx+b que foi reconfigurada para ter um comportamento logarítmico.

À medida que vão sendo registadas as tensões de saída, vão sendo processadas pelo código e inseridas nas suas curvas logarítmicas. Seria esperado que os valores seguissem minimamente o comportamento das calibrações de fábrica como se observa nas figuras 4.2.1. Das diversas tentativas executadas no MATLAB para tentar recriar uma curva que fizesse sentido, nenhuma satisfez quando comparadas com as medições registadas no Telaire. O que se sucede é que é extremamente complicado fazer uma medição da tensão de saída, seja pelo Arduíno IDE, seja pelo multímetro que não contenha ruído ou que não seja diretamente influenciada por humidade, mas especialmente, pela temperatura provocando erro ou pelo menos dúvida quanto à veracidade das medições.

4.2.2 Calibração por Método dos Mínimos Quadrados

Este método foi o predileto não só pela sua simplicidade, mas também pela sua robustez face aos métodos supramencionados, facto que se tornou cada vez mais claro ao realizar as experiências. Encontrar a reta que melhor se adequa a um determinado set de dados é uma das melhores formas

de se combater a incerteza associada a uma calibração. Existem diversos métodos disponíveis para se encontrar a melhor curva de ajuste aos dados que foram obtidos mas, para todos os efeitos, as rotinas do método dos mínimos quadrados com regressão linear são bastante boas e amplamente utilizados na literatura científica.

A regressão linear é um método matemático estatístico que visa procurar estabelecer uma correlação entre duas variáveis sendo que, neste caso, tenciona-se correlacionar o valor de tensão dos sensores low cost com o valor de ppm dos sensores mais dispendiosos. O objetivo, se bem-sucedido, passa por criar um modelo de previsão para que quando se obtenha um determinado valor de tensão com os sensores low cost, sabermos aproximadamente a quantos ppm reais este corresponde.
Os códigos do Arduíno IDE são também mais fáceis de programar, pois apenas necessitam de conhecimentos básicos de C++ quando comparados com o método de calibração de fábrica, que se não existisse um código proposto pelo fabricante, seria extremamente complicado criá-lo de raiz. Só precisamos do output de tensão de saída do sensor fornecido pelo Arduíno e dos dados do sensor dispendioso medidos ao mesmo tempo, na mesma localização, e alocar esses valores num ficheiro de modo a poder analisá-los, um versus o outro, num gráfico XY.

Capítulo 5

Resultados

No capítulo presente não consta tudo o que concerne a resultados da construção do protótipo e a toda a situação de montagem e funcionamento dos componentes, porque se considerou que deveriam ser referidos apenas nas conclusões.

Neste capítulo serão colocadas as bases que possibilitam retirar conclusões acerca da performance dos sensores, mas não foi o foco principal da dissertação devido a constrangimentos de tempo contudo é um constituinte de importância para futuras investigações.

5.1 Resultados das Calibrações

Foram executados diversos testes que foram aglomerados e catalogados como sendo 6 experiências distintas. Estas foram realizadas com o intuito de averiguar a performance dos sensores MG-811 e MQ-131.

Nesta secção quando se menciona pré-aquecimento, estamos a fazer referência a um período em horas, onde o sensor que previamente estava desligado, é agora ligado à corrente elétrica durante esse determinado período de tempo antes de se começar efetivamente a fazer medições de calibração. Seguidamente demonstram-se os resultados fazendo pequenas introduções ou explicações acerca do que estamos a observar.

Todas as equações nos gráficos representados nesta secção estão resolvidos em ordem à tensão, porque descrevem a equação da reta que melhor se ajusta ao gráfico. Para se obter a equação que irá fazer a transformação de tensão para ppm no Arduíno devemos manipular a equação 5.2 como demonstrado.

$$\begin{aligned} V &= m \times ppm + b \\ ppm &= \frac{V - b}{m} \end{aligned} \tag{5.1}$$

5.1.1 MG-811

Como o MG-811 foi um dos primeiros sensores a ser colocado em funcionamento de forma bem sucedida, iniciaram-se as calibrações com este sensor em foco. Foram concretizadas 5 tentativas para calibrar o sensor. Estas tentativas estão intrinsecamente relacionadas com diferentes fases da construção e despiste dos problemas no protótipo.

Na primeira tentativa, experimentou-se calibrar o sensor com alimentação vinda do Arduíno a 5 V como mencionado na pré-montagem do protótipo. Foram aprovisionadas apenas duas horas de pré-aquecimento para se testar o comportamento do sensor e aferir através de futuras tentativas, se realmente um pré-aquecimento mais longo faria ou não diferença ao sensor para atingir o seu ponto químico de equilíbrio.

A figura gerada em MATLAB pelo conjunto de dados recolhidos do MG-811 versus o Telaire é o seguinte:

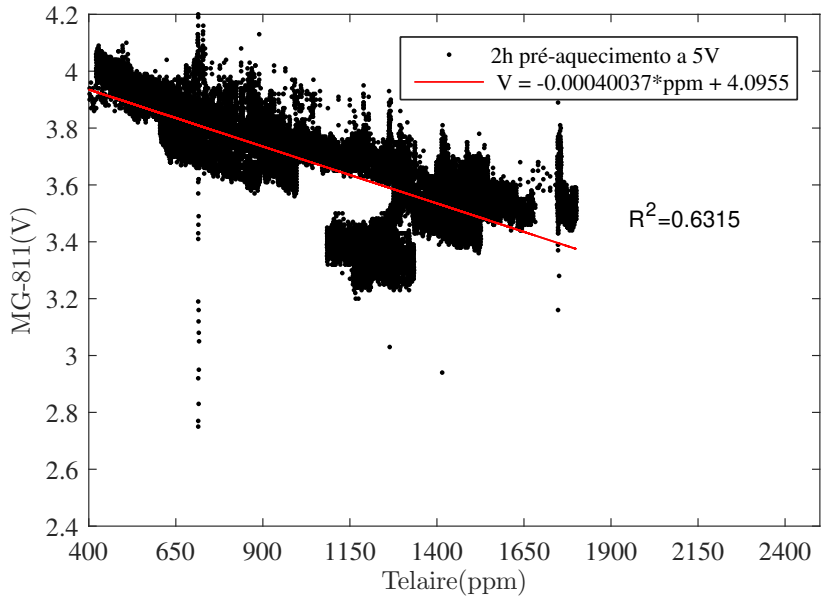


Figura 5.1.1: Calibração MG-811 vs. Telaire - 1ª Tentativa

Podemos observar na figura 5.1.1 da primeira tentativa, que parece existir um comportamento estranho em determinadas secções das medições efetuadas pelo sensor e recolhidas pelo Arduíno. Ao início foi problemático tentar compreender o que se passava e estes resultados não se assumiam promissores. No entanto, de forma mais minuciosa observou-se que determinados pontos não pertencem a um seguimento lógico de medições, como por exemplo, os valores registados entre os 650 e 900 ppm entre os 2.7 e 3.6 V de tensão. Fomos levados a crer que algo não estava certo e consequentemente tomaram-se algumas decisões para a segunda tentativa.

Na segunda tentativa, como referenciado na montagem do protótipo alterou-se a posição do MG-811 dentro da caixa, mas foi continuada a alimentação ao sensor de 5 V através do Arduíno e

experimentou-se também um pré-aquecimento de 4 horas em vez de 2 horas, para se tentar despistar o problema dos pontos com um comportamento invulgar.

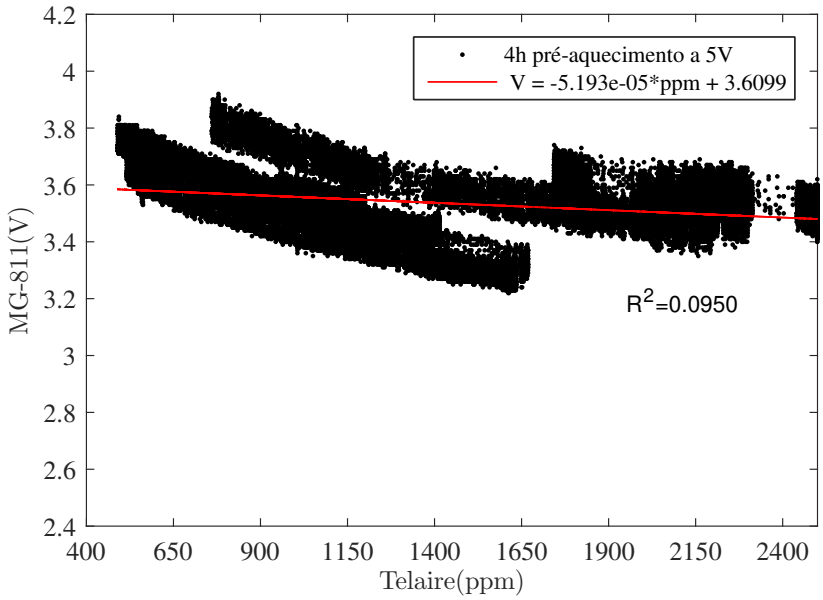


Figura 5.1.2: Calibração MG-811 vs. Telaire - 2ª Tentativa

Nesta segunda iteração representada na figura 5.1.2 com a posição do MG-811 dentro da caixa alterada, deixamos de ter dados completamente ilógicos como na primeira tentativa 5.1.1, porém podemos notar que existe uma nuvem de pontos na parte superior do gráfico e uma na parte inferior e aparenta existir um deslocamento dos pontos para a direita. Relembramos que embora o gráfico pareça ser contínuo, não é nada mais que uma aglomeração de medições de 4 horas. A nuvem de baixo corresponde às 20 primeiras horas de medições que foram efetuadas logo após o pré-aquecimento. A nuvem superior são as medições efetuadas mais tarde, correspondendo a 12 horas de medições. Existem duas teorias para tentar explicar este comportamento do deslocamento das nuvens de pontos: uma é que o sensor efetivamente precisa de muito mais horas de pré-aquecimento para atingir o seu equilíbrio químico e enquanto não o atinge, vão existir alterações nas medições dos valores de tensão de saída; A outra hipótese é a de se ter verificado que ao fim de um tempo indeterminado, o potenciómetro do LM2596S que traduz a tensão do transformador para o Arduíno, apresentava valores diferentes daqueles para os quais tinha sido inicialmente configurado. Deu-se conta deste facto enquanto se verificava a montagem entre as experiências, onde se reparou que em vez de estar a fornecer 6 V ao Arduíno, estava a fornecer 6.4 V. Um incremento de 0.4 V na tensão de alimentação, pode-se traduzir na maior tensão de saída registada a elevados ppm na nuvem superior. Numa situação normal seria esperado que quanto maior a concentração de gás, menor o valor da tensão de saída, precisamente o contrário do que se observa na figura 5.1.2.

No decorrer da terceira tentativa de calibração foi o período de reparar as falhas da segunda tentativa. As falhas da experiência anterior foram combatidas ao decidir-se que se ia testar um pré-

aquecimento mais longo de 24 horas. Foi decidido também que iria ser verificada regularmente a tensão de saída do LM2596S para o Arduíno de modo a que durante a experiência toda se mantivesse a 6 V, com o intuito de garantir que a tensão recebida e fornecida pelo Arduíno é constante. A representação gráfica da terceira tentativa encontra-se na figura 5.1.3.

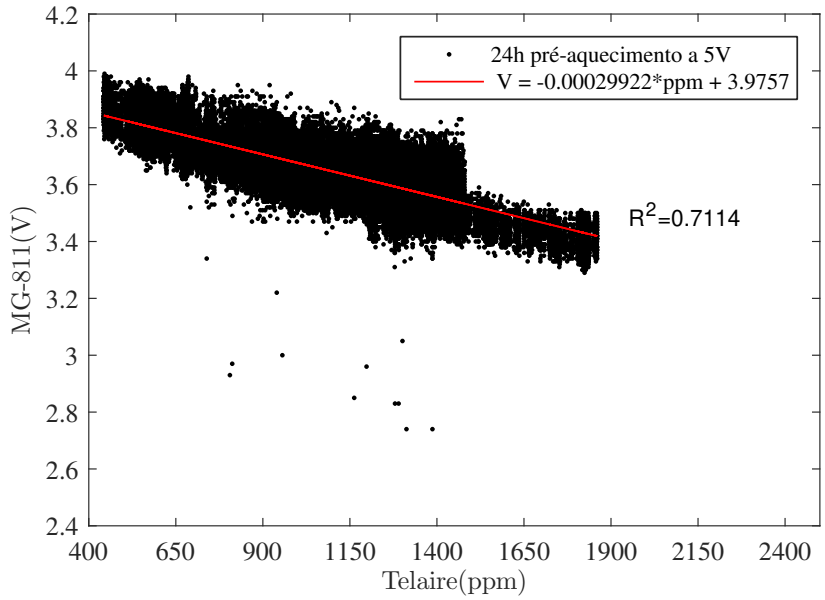


Figura 5.1.3: Calibração MG-811 vs. Telaire - 3ª Tentativa

Ao observar a figura 5.1.3 podemos notar imediatamente que foi corrigido o efeito das nuvens de valores completamente desfasadas. Podemos argumentar que manter a tensão fixa e ter um bom período de pré-aquecimento, ajuda a corrigir o desfasamento de tensões de saída de medições a horas diferentes. Apesar do gráfico assumir um comportamento mais razoável em relação aquilo que seria esperado, ainda existem uns valores de tensão de saída completamente ilógicos entre os 650 e 1400 ppm e 2.6 e 3.4 V.

Fazer o despiste desta situação foi possível quando se constatou, que a luz do potenciómetro acendia durante algumas instâncias em que se preparava a tentativa de calibração seguinte. Depreendeu-se que algo não estava certo e anotou-se a hora, minuto e segundo de um desses acontecimentos. Ao analisar os ficheiros Excel correspondentes a esse determinado período de tempo, foi possível ver que o facto da luz acender, estava diretamente relacionado com uma quebra de tensão. É possível ver este comportamento na figura analisando os pontos ilógicos entre os 700 e 1400 ppm e os 2.6 e 3.3 V.

Como plano de ação para a tentativa 4, foi decidido que se ia prolongar ainda mais o período de pré-aquecimento, desta vez até às recomendadas 48 horas pelo datasheet. Foi também tomada uma decisão no que diz respeito à alimentação do sensor de modo a tentar corrigir os pontos com comportamento estranho, decisão essa que passou por alimentar os sensores diretamente com 6V em vez de utilizar o Arduíno para os alimentar a 5 V. A razão que nos levou a tal reside no facto de que ao manipular o potenciómetro do LM2596S e testar diferentes configurações de distribuição

de tensão, observou-se que a luz do potenciómetro do MG-811 desligava assim que se fornecia mais tensão. Esta decisão de se fornecer diretamente os 6 V aos sensores não foi a mais correta e como mencionado no capítulo da construção, nasceu da ignorância dos limites de tensão e corrente do Arduíno. Os resultados destas decisões podem ser observados na figura 5.1.4.

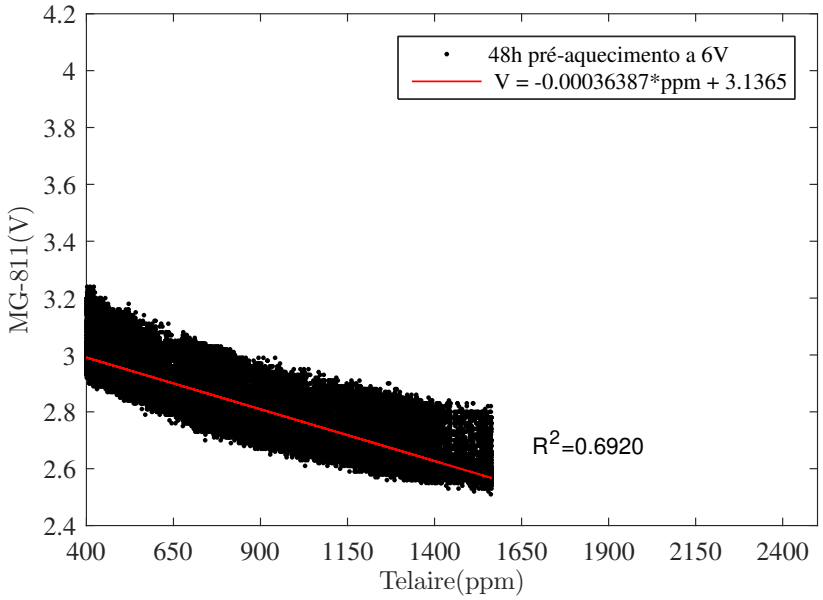


Figura 5.1.4: Calibração MG-811 vs. Telaire - 4ª Tentativa

Apesar de algumas decisões erradas durante estas tentativas, o que é certo é que o comportamento do gráfico na tentativa número 4 começou a aproximar-se de um comportamento mais adequado. Deixamos de ter nuvens de pontos desfasadas e deixamos de ter pontos ilógicos. Apesar de tudo denota-se uma grande variância nos dados recolhidos. O R^2 considera-se satisfatório se tivermos em conta o preço do sensor assim como a total desconsideração da inclusão da temperatura e humidade nas análises.

Podemos ainda argumentar que das 24 horas de pré-aquecimento até às 48 horas não parece existir grande diferença, sendo que manter uma tensão constante é o fator que se julga ser mais importante.

A última experiência, presente na figura 5.1.5, serviu como modo de confirmar que é possível recrear a experiência com alguma fiabilidade e argumentar que existem indícios de que talvez seja possível efetuar bons testes de repetibilidade no futuro, pois o sensor ao usufruir de 48 horas de pré-aquecimento e tensão e corrente elétrica constante, parece comportar-se de uma forma mais satisfatória.

Notamos pelo gráfico que o seu comportamento é bastante semelhante ao da tentativa 4 representada na figura 5.1.4. O seu R^2 é menor devido à sua maior variação, no entanto, podemos argumentar que a sua trend line e equação continua a fazer sentido não destoando muito da equação da tentativa 4. Relembramos nesta parte final da análise de resultados, que não foi possível

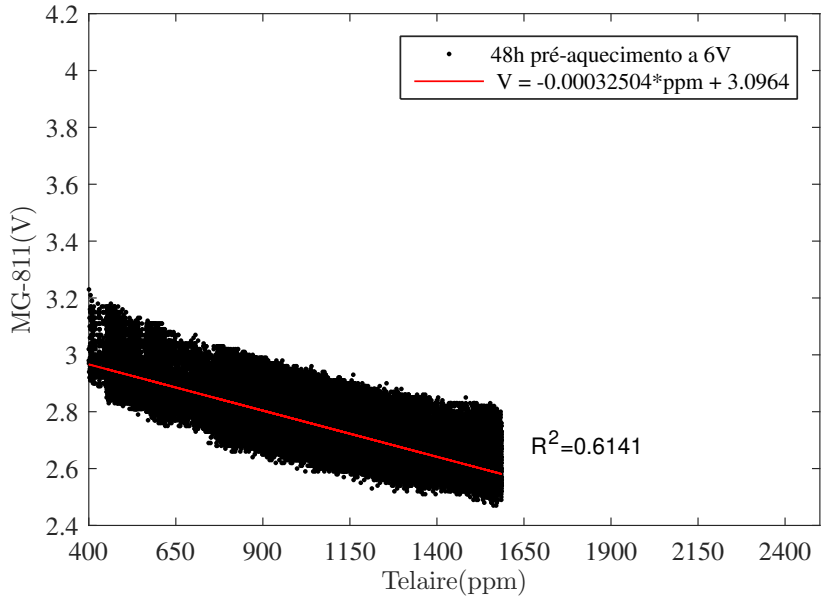


Figura 5.1.5: Calibração MG-811 vs. Telaire - 5ª Tentativa

ter controlo sobre a temperatura e humidade relativa que podem, juntamente com todas as outras nuances, explicar grande parte das diversas variâncias de valores entre tentativas.

5.1.2 MQ-131

Começamos a secção referente ao MQ-131 por dizer que infelizmente não foi possível realizar mais do que uma tentativa que se considerasse útil.

Após se ter conseguido as tentativas 4 e 5 do MG-811, consideradas satisfatórias, aplicaram-se esses conhecimentos ao MQ-131. Foi garantida uma tensão de 6 V constante, diretamente vinda do transformador e foi dado um pré-aquecimento de 48 horas. Na figura 5.1.6 podemos observar os resultados.
Denotamos no gráfico um comportamento onde o espaçamento horizontal entre cada ponto de dados é sempre igual, este advém da resolução do sensor Aeroqual 500 que faz medições de ozono de 1 em 1 $\mu g\,m^{-3}$. Como estamos a falar de um conjunto de números naturais que posteriormente é multiplicado por uma constante que os transforma em ppm, logicamente mantemos um espaçamento fixo entre medições que é o que podemos observar no gráfico.

Curiosamente, apesar de alguns valores com comportamento estranho similar às tentativas iniciais do MG-811, obtivemos a maior correlação de dados com um R^2 de 0.88. Porém não se considera que este modelo seja satisfatório por uma razão simples, não foram elaboradas tentativas suficientes de calibração que ajudassem a despistar a razão para termos dados que aparentam sair fora da norma ou que não seguem uma tendência. Como não podemos afirmar ou desmentir pura aleatoriedade, considera-se esta calibração sem efeito.

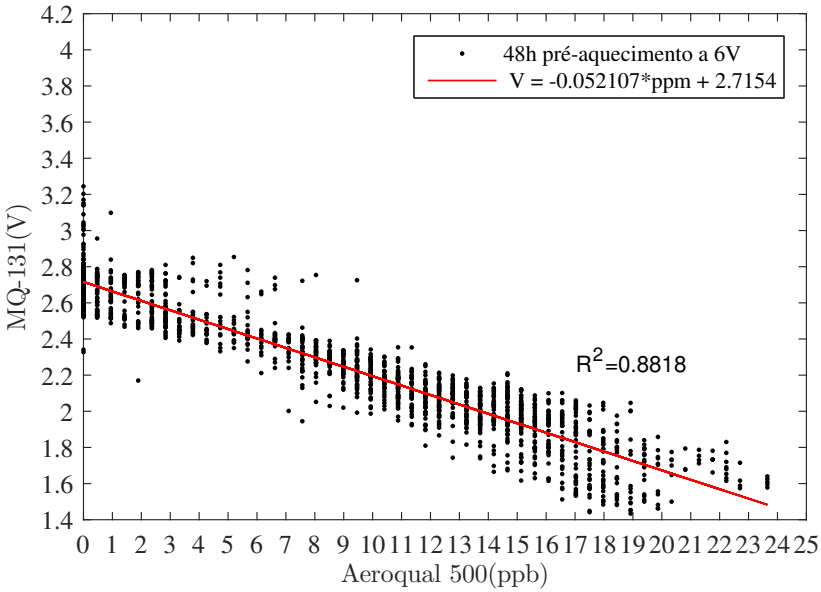


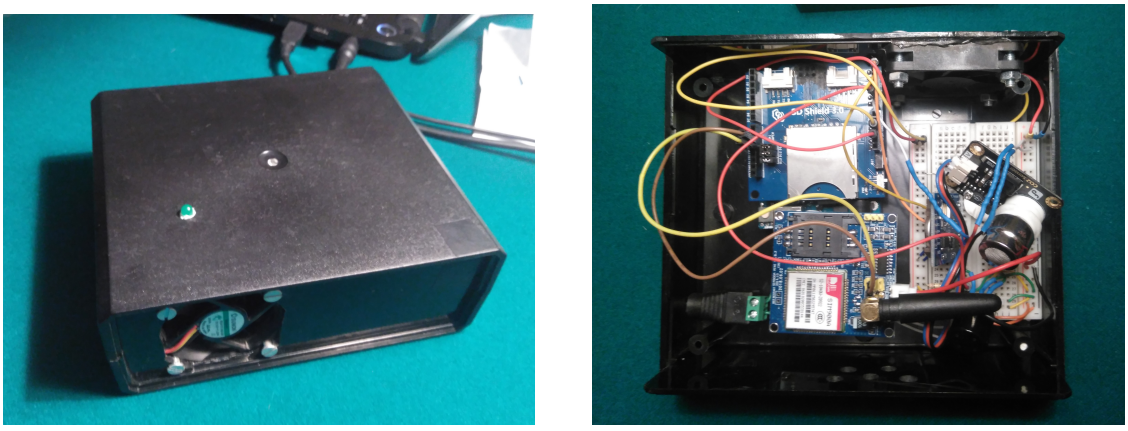
Figura 5.1.6: Calibração MQ-131 vs. Telaire - 1ª Tentativa

Capítulo 6

Conclusões e Trabalho Futuro

Neste capitulo serão retiradas conclusões acerca das fases mais importantes do desenvolvimento e teste do protótipo desenvolvido.

Começando pela própria caixa do protótipo e nas aferições sobre o que diz respeito à sua construção, podemos notar pela figura 6.0.1 que com todos os componentes montados existe espaço para melhorias. É possível inferir que tanto as suas dimensões como o seu peso de cerca de 442 g podem ser reduzidos. Existe bastante espaço horizontal e vertical inutilizado dentro da caixa pelo que seria interessante em futuros projetos fazer o invólucro em impressora 3D, sendo assim possível fazer uma caixa que vá exatamente ao encontro das necessidades.



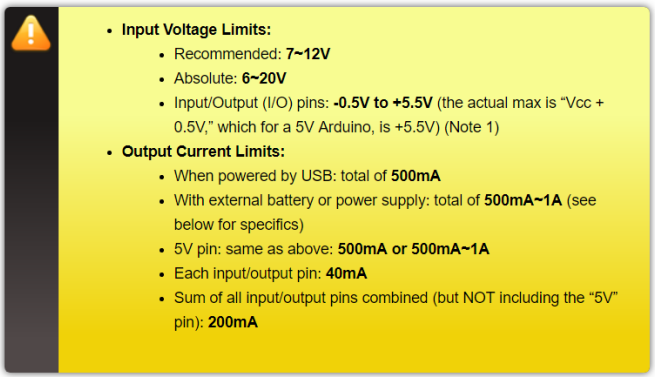
(a) Caixa fechada

(b) Caixa aberta

Figura 6.0.1: Imagem do protótipo

Outro aspeto relevante que possibilita ainda mais a redução das dimensões do protótipo, é referente à utilização de circuitos PCB ou circuitos integrados, pois podemos eliminar por completo material supérfluo como cabos ou resistências de grandes dimensões. Destes dois circuitos aconselha-se o circuito integrado, pois têm menos ruído, alta performance, consomem menos energia e são extremamente fiáveis.

Quanto ao esquema da distribuição de tensão do protótipo é sabido que passou por diversas iterações à medida que se foi progredindo na construção e teste do protótipo. Um dos erros mais importantes que devemos admitir e corrigir em iterações futuras está relacionado com o mau planeamento da distribuição de tensão do transformador para o Arduíno e os sensores. Na tabela seguinte podemos observar os limites do Arduíno:



- **Input Voltage Limits:**
  - Recommended: **7~12V**
  - Absolute: **6~20V**
  - Input/Output (I/O) pins: **-0.5V to +5.5V** (the actual max is "Vcc + 0.5V," which for a 5V Arduino, is +5.5V) (Note 1)
- **Output Current Limits:**
  - When powered by USB: total of **500mA**
  - With external battery or power supply: total of **500mA~1A** (see below for specifics)
  - 5V pin: same as above: **500mA or 500mA~1A**
  - Each input/output pin: **40mA**
  - Sum of all input/output pins combined (but NOT including the "5V" pin): **200mA**

Figura 6.0.2: Limites de tensão e corrente do Arduíno

A partir daqui podemos concluir com um bom grau de certeza que o mau funcionamento do MG-811 durante as experiências, quando este estava a ser alimentado pelo Arduíno a 5 V, teve a sua origem na fraca alimentação de tensão e corrente elétrica ao Arduíno. Podemos constatar na figura 6.0.2 que o Arduíno tem um limite de corrente de saída de 1 A em condições máximas quando alimentado pela ligação DC a pelo menos 7 V. É importante frisar que isto são condições máximas que provavelmente não representam a realidade, o normal é o Arduíno conseguir fornecer cerca de 400-800 mA. Olhando para a tabela 3.2.2 onde estão descriminados os consumos de corrente dos componentes, podemos entender que é possível que o Arduíno não estivesse a lidar bem com a carga que tinha de distribuir e pode muito bem explicar o porque de depois de se ter mudado a fonte de alimentação do MG-811 para ser diretamente alimentado pelo transformador, o problema tenha parado. Mais ainda, nunca foram fornecidos 7 V ao Arduíno mas sim, como foi discutido ao longo da dissertação, um máximo de 6 V. Outra hipótese considerada que pode ou não ter tido também algum impacto, é EMI (*Electromagnetic Interference*), como temos uma ventoinha, temos um motor, um motor pode gerar um campo eletromagnético que interfere com os componentes ao criar ruído. A razão para se acreditar que tal possa ter ocorrido é devido a alguns acontecimentos esporádicos onde ao alterar a posição do MG-811 na caixa, por vezes a luz do potenciómetro desligava, levando a crer que existiam influencias externas que não unicamente a corrente de saída do Arduíno.

Concluímos a discussão da tensão e corrente elétrica aconselhando que para se evitar futuros erros convém fornecer ao Arduíno Uno pelo menos 8 V pois este tem uma queda de tensão de 0.5 a 0.7 V quando passa pelo regulador de tensão interno do Arduíno e como queremos garantir o seu bom funcionamento devemos garantir uma tensão elevada o suficiente para tal[45].

Quanto ao resto do circuito e com o Arduíno a funcionar em pleno, não deverá ser problema este

alimentar o MG-811. Porém, devemos implementar um LM2596S extra porque não devemos sobrecarregar o Arduíno com sensores que são bastante consumistas de corrente, como é o caso do MQ-131, deste modo, podemos deixar o Arduíno uno a alimentar alguns dos componentes com menos consumo e utilizamos o outro LM2596S para converter os 12 V do transformador em 5 V e alimentar diretamente o sensor MQ-131. Tendo em conta que a bateria utilizada consegue fornecer até 2 A e o circuito todo consome no máximo 1.8 A, estamos seguros de que é possível no futuro fazer calibrações com um maior grau de fiabilidade.

Nos processos de calibração, existiram diversos pormenores que tiveram de ser bem ponderados e talvez o mais importante esteve relacionado com o offset de dados e o facto do código do Arduíno criar uma falha de 1 segundo. A razão porque não se considerou problemática esta situação teve como base o seguinte pensamento: o processamento por parte do Arduíno está a demorar um terminado período de tempo fixo, ou seja, cada vez que o código percorre o ciclo, o Arduíno não consegue debitar os dados referentes à tensão de saída do sensor com o atraso incutido de 1000 ms, mas consegue, por exemplo, debitar esses registos de 1.23 em 1.23 segundos, demorando os extra 0.23 segundos que adiciona ao atraso incutido no código. O que se quer dizer com isto é que o espaçamento entre medições é fixo na sua continuidade e sendo o espaçamento fixo e sabendo o inicio e fim das medições, temos 2 vetores que são equivalentes. Com estas medições equivalentes basta uma simples interpolação em MATLAB para os transformar no mesmo tamanho, o que deverá manter o mesmo nível de credibilidade para serem comparados um com o outro. No futuro, para contornar este problema deve-se reduzir a utilização de strings e optar por char, arranjando maneira de fazer os timestamps de uma forma mais leve para que as linhas de código não tenham de realizar operações redundantes para chegar ao mesmo objetivo final.

Referente aos sensores adquiridos para esta dissertação, como havia já sido advertido, as expetativas de performance destes sensores não eram as melhores, não só dado ao seu preço extremamente low cost como a todos os problemas que foram encontrados no desenrolar da montagem e das experiências. Apesar de tudo, o MG-811 obteve na tentativa 4 e 5 um comportamento aceitável neste contexto. A ilação que se retirou destas experiências de calibração foi que estes sensores, supondo que são calibrados sem se ter em conta as variáveis da temperatura e humidade, não são bons o suficiente para medir com uma precisão de 2, 20 ou até 50 ppm, mas são bons o suficiente para detetar uma variação na qualidade do ar.

Foi ponderada a utilização dos plots residuais para tentar validar os dados, no entanto não fazia sentido utilizar os plots residuais de milhares de dados pois até um gráfico que aparentasse ter aleatoriedade, se continuássemos a apresentar cada vez mais dados até à ordem dos milhares, eventualmente iria formar um padrão. Queremos com isto dizer que os plots residuais de todos os conjuntos de dados destas experiências aparentavam ter um padrão bem distinto de uma mancha de pontos, o que por norma leva a crer que existe uma variável que não foi contabilizada. Esta conclusão parece-nos elementar pois sabemos efetivamente que existem duas variáveis não contabilizadas: a temperatura e a humidade relativa. Por estas razões não foram incluídos esses gráficos na análise de resultados.

Por último, em relação aos sensores utilizados, aconselha-se a utilizar um sensor NDIR que é teoricamente mais preciso e menos consumidor de energia do que o MG-811. Um exemplo de um tipo de sensor NDIR low cost medidor de CO_2 é o do MH-Z14. Posteriormente poderá também ser incluído um sensor de NO_x ou SO_x da gama MQ.

Anexo A

ANEXO

A.1 CÓDIGOS UTILIZADOS

```
//-------------------------------------------------------------
//Codigo do prototipo compilado e modificado por:
//ANGELO SOARES
//Faculdade de ciencias 2018

//Inclusao de todas as libraries necessarias
#include <SoftwareSerial.h>
#include <String.h>
#include <dht.h>
dht DHT;  //Iniciar o DHT
#define DHT11_PIN 6 //definir o pin que o DHT usa
SoftwareSerial mySerial(10, 11);  //iniciar

boolean pin2=LOW, pin3=LOW, pin4=LOW, pin5=LOW, pin6=LOW;  //Garante que o sinal do
    Pin comeca LOW

void setup()
{
  mySerial.begin(9600);  // the GPRS baud rate
  Serial.begin(9600);    // the GPRS baud rate
  pinMode(2,INPUT);
  pinMode(3,INPUT);
  pinMode(4,INPUT);
  pinMode(5,INPUT);
  pinMode(6,INPUT);
  delay(1000);
}
void loop()
{
  Send2Internet();

  if (mySerial.available())
```

```
  Serial.write(mySerial.read());
  delay(1000);
}
void Send2Internet()
{
  int O3 = analogRead(A0);
  int CO2 = analogRead(A1);
  int chk = DHT.read11(DHT11_PIN);

  float voltageO3 = O3 * (5.0 / 1023.0);
  float voltageCO2 = CO2 * (5.0 / 1023.0);
  mySerial.println("AT");
  delay(1000);

  mySerial.println("AT+CPIN?");    //Alguns destes comandos sao apenas de
    seguranca,
  //ou teste para se saber se tudo esta ok com a conexao, sao bons para
    despistar qualquer problema
  //antes de se comecar a enviar dados.
  delay(1000);

  mySerial.println("AT+CREG?");
  delay(1000);

  mySerial.println("AT+CGATT?");
  delay(1000);

  mySerial.println("AT+CIPSHUT");
  delay(1000);

  mySerial.println("AT+CIPSTATUS");
  delay(2000);

  mySerial.println("AT+CIPMUX=0");
  delay(2000);

  ShowSerialData();

  mySerial.println("AT+CSTT=\"net2.vodafone.pt\",\"vodafone\",\"vodafone\"");
  //configuracao APN da vodafone, depende do operador. "NOME" , "Utilizador" ,
    "Password"
  delay(1000);

  ShowSerialData();

  mySerial.println("AT+CIICR");  //Iniciar a conexao wireless
  delay(3000);

  ShowSerialData();
```

```
79  mySerial.println("AT+CIFSR");  //Obter o IP local
    delay(2000);
81
    ShowSerialData();
83
    mySerial.println("AT+CIPSPRT=0");
85  delay(3000);

87  ShowSerialData();

89  mySerial.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",\"80\"");
    //Iniciar a conexao com o api do thingspeak  "TIPO DE COMUNICACAO", "API" ,
91  //"PORTA" :  normalmente e a <80> mesmo em outros APIs
    delay(6000);
93
    ShowSerialData();
95
    mySerial.println("AT+CIPSEND");  //Iniciar o modo de enviar dados
97  delay(4000);
    ShowSerialData();
99
    String str="GET /update?api_key=CMXQSMEW2AN544JF&field1=" + String(voltageO3)
      + "&field2=" + String(voltageCO2);
101 //O string de dados que vai ser enviado para o APN <CMXQSMEW2AN544JF> +
      String de dados
    mySerial.println(str); //envia os dados no <String str>
103 ShowSerialData();
    delay(4000);
105
    //para se mandar tambem info da temperatura e humidade devemos substituir o
      String str com a seguinte
107 //linha de codigo:
    //String str="GET /update?api_key=CMXQSMEW2AN544JF&field1=" + String(
      voltageO3) + "&field2="
109 //+ String(voltageCO2) + "&field3=" + String(DHT.temperature) + "&field4=" +
      String(DHT.humidity);  ;

111 mySerial.println((char)26);
    //A enviar a informacao. (char)26 representa um character ASCII que neste
      codigo indica quando e o fim do string.
113 //em vez de manualmente dizermos que o string vai ter 20 characters
    // colocamos o char 26 para indicar que ele deve confirmar o envio quando
      receber o ultimo character,
115 //seja qual for o tamanho do string.

117 delay(5000);  //esperando a resposta de volta a dizer que foi confirmado.
    //Devemos colocar um bom delay pois esta confirmacao depende das condicoes da
      internet
```

```
119   //nao e comum demorar muito tempo, 5s e seguro.
      mySerial.println();
121
      ShowSerialData();
123
      mySerial.println("AT+CIPSHUT");   //fecha a conexao e seguidamente o loop
        volta ao inicio reiniciando o processo.
125   delay(100000);
      ShowSerialData();
127
}
129 void ShowSerialData()     //funcao que mostra no seria monitor do IDE arduino
      tudo a acontecer
//caso o Arduino esteja ligado a um computador.
131
{
133   while(mySerial.available()!=0)
      Serial.write(mySerial.read());
135 }
```

Listing A.1: Código para fazer a comunicação entre o Arduíno e sensores com o website ThingSpeak

```
//-------------------------------------------------------------
//Codigo do prototipo compilado e modificado por:
//ANGELO SOARES
//Faculdade de ciencias 2018

//Inclusao de todas as libraries necessarias
#include <SD.h>
#include <SPI.h>
#include <dht.h>
//#include <hpma115S0.h>
#include "Arduino.h"
#include <SoftwareSerial.h>
#include <Wire.h>
#include <TimeLib.h>
#include <DS1307RTC.h>

//USAR RTC PINS SCL SDA - A5 e A4 respetivamente
dht DHT;

const int CS_pin = 4;
int SDPIN = 8;
long id = 1;

//Descomentar caso se queira utilizar HPMA S
//SoftwareSerial hpmaSerial(5, 3); // Tx Rx PM sensor
//Descomentar caso se queira utilizar HPMA
//HPMA115S0 hpma115S0(hpmaSerial);

#define DHT11_PIN 6
char timeStamp[10];
char dateStamp[10];
File dataFile;

//Iniciar o SD card e o HPMA
void setup()
{
  Serial.begin(9600);
  Serial.println("Initializing Card");
  //CS Pin is an output
  pinMode(CS_pin, OUTPUT);

//  //Nao foi utilizado o sensor PM mas fica aqui o codigo. Begin PM sensor
//  hpmaSerial.begin(9600);
//  delay (3000);
//  hpma115S0.Init();
//  hpma115S0.StartParticleMeasurement();

  //o SD shield vai buscar power ao pin 8 (Nao utilizar um sensor conectado ao
    pin 8)
```

```
49    pinMode(SDPIN, OUTPUT);
      digitalWrite(SDPIN, HIGH);
51
      if (!SD.begin(CS_pin))
53    {
        Serial.println("Card Failure");
55      return;
      }
57    Serial.println("Card Ready");
      File dataFile = SD.open("LOG.csv", FILE_WRITE);
59    if (dataFile)
      {
61      dataFile.println(", , , ,"); //Uma linha em branco para separar sets de
        dados diferentes
        String header = "Time,     Date,     ID, Temp,  HR,  O3, CO2";
63      dataFile.println(header);
        dataFile.close();
65      Serial.println(header);
      }
67    else
      {
69      Serial.println("Couldn't open log file");
      }
71  }

73  //Iniciar o loop de leitura de todas as variaveis e guardar no cartao SD
    //com o respetivo timestamp
75  void loop() {

77    tmElements_t tm; //iniciar comandos RTC, ele vai buscar DATA e Hora atraves
        do comando "tm"
      int chk = DHT.read11(DHT11_PIN);
79  //  unsigned int pm2_5, pm10;
      int O3 = analogRead(A0);
81    int CO2 = analogRead(A1);

83    float voltageO3 = O3 * (5.0 / 1023.0);
      float voltageCO2 = CO2 * (5.0 / 1023.0);
85
      if (RTC.read(tm)) {
87      sprintf(timeStamp, "%02d:%02d:%02d", tm.Hour, tm.Minute, tm.Second);
        //Serial.println(timeStamp);
89      sprintf(dateStamp, "%2d/%2d/%2d", tm.Day, tm.Month, tm.Year);
        //Serial.println(dateStamp);
91    }

93  //hpma115S0.ReadParticleMeasurement(&pm2_5, &pm10);
    //string com todos os dados que este deve guardar no SD card.
```

```
95    String dataString = String(timeStamp) + ", " + String(dateStamp) + ", " +
        String(id) + ", " + String(DHT.temperature)
      + ", " + String(DHT.humidity) + ",  " + String(voltageO3) + ", " + String(
        voltageCO2);
97    File dataFile = SD.open("LOG.csv", FILE_WRITE);

99    if (dataFile)
      {
101     dataFile.println(dataString);
        dataFile.close();
103     Serial.println(dataString);
      }
105   else
      {
107     Serial.println("Couldn't open log file");
      }
109   id++;
      dataFile.close();
111   delay(1000);
}
```

Listing A.2: Código para se fazer a comunicação entre o Arduíno e sensores com o SD Shield 3.0

```
//Codigo de teste analogico simples - Retirado da pagina do Arduino
//https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage

void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0); //Sensor conectado ao pin A0
  float voltage= sensorValue * (5.0 / 1023.0);
  Serial.println(voltage)
}
```

Listing A.3: Código simples para testar a tensão de saída dos sensores com o Arduíno

```
%Codigo MATLAB elaborado por Angelo Soares
%Faculdade de Ciencias 2018

%Codigo que tem de ser utilizado com o export_fig e o ghostscript
%instalados

clc; clear all; close all

%Fazer a chamada das folhas Excel .csv
EXCEL_A=xlsread('CO2 8 de Agosto 00-04.csv',1);
EXCEL_B=xlsread('CO2 8 de Agosto 11-15.csv',1);
EXCEL_C=xlsread('CO2 8 de Agosto 14-18.csv',1);
EXCEL_D=xlsread('CO2 8 de Agosto 19-23.csv',1);
EXCEL_E=xlsread('CO2 9 de Agosto 1-5.csv',1);
EXCEL_F=xlsread('CO2 9 de Agosto 10-14.csv',1);

%Iniciar os vetores de dados, A sao do telaire, AA sao do sensor MG-811
%neste caso
A = EXCEL_A([3:14466],5);
AA = EXCEL_A([3:12564],6);
B = EXCEL_B([3:14466],5);
BB = EXCEL_B([3:12552],6);
C = EXCEL_C([3:14466],5);
CC = EXCEL_C([3:12530],6);
D = EXCEL_D([3:14466],5);
DD = EXCEL_D([3:12520],6);
E = EXCEL_E([3:14466],5);
EE = EXCEL_E([3:12515],6);
F = EXCEL_F([3:14466],5);
FF = EXCEL_F([3:12473],6);

%Transformar o maior vetor no tamanho do mais pequeno atraves de
%interpolacao
A1 = interp1(1:length(A),A,linspace(1,length(A),length(AA)));%Encolher A para
    caber em B
A1 = A1';
B1 = interp1(1:length(B),B,linspace(1,length(B),length(BB)));
B1 = B1';
C1 = interp1(1:length(C),C,linspace(1,length(C),length(CC)));
C1 = C1';
D1 = interp1(1:length(D),D,linspace(1,length(D),length(DD)));
D1 = D1';
E1 = interp1(1:length(E),E,linspace(1,length(E),length(EE)));
E1 = E1';
F1 = interp1(1:length(F),F,linspace(1,length(F),length(FF)));
F1 = F1';

%concatenar os vetores num so
telaire = cat(1,A1,B1,C1,D1,E1,F1);
```

```
myco2 = cat(1,AA,BB,CC,DD,EE,FF);

%Propriedados do grafico para serem demonstrados em latex
figure('Units','inches',...
    'PaperPositionMode','auto');

%Scatter dos dados e consequente demonstracao da equacao que melhor se
%ajusta aos dados, assim como display do seu R2
scatter(telaire,myco2,'.','k');
P = polyfit(telaire,myco2,1);
yfit = polyval(P,telaire);
hold on;
box on;
plot(telaire,yfit,'r');
a=P(1);
b=P(2);
str = [' V = ' num2str(a) '*ppm + ' num2str(b)];
mdl1=LinearModel.fit(telaire,myco2);
R= mdl1.Rsquared.Ordinary;

txt1 = 'R^{2}=0.6920';
text(1680,2.7,txt1)

%Propriedados do grafico para serem demonstrados em latex
xlabel({'Telaire(ppm)'},...
'FontUnits','points',...
'interpreter','latex',...
'FontSize',11,...
'FontName','Times')
ylabel({'MG-811(V)'},...
'FontUnits','points',...
'interpreter','latex',...
'FontSize',11,...
'FontName','Times')
axis([400 2500 2.4 4.2])
set(gca,...
'Units','normalized',...
'YTick',2.4:.2:4.2,...
'XTick',400:250:2500,...
'Position',[.15 .2 .75 .7],...
'FontUnits','points',...
'FontWeight','normal',...
'FontSize',11,...
'FontName','Times')

%Propriedados do grafico para serem demonstrados em latex
legend('48h pre-aquecimento a 6V', str,  ...
'FontUnits','points',...
'interpreter','latex','FontSize',11,...
```

```
'FontName','Times','Location','NorthEast');

set(gcf, 'Color', 'w');
export_fig -pdf
```

Listing A.4: Código de MATLAB utilizado para as computações dos resultados

```
//Codigo de Davide Girono para calibrar os sensores MQ com metodos alternativos
//sensor input PIN
int mqInput = A1;
//pull-down resistor value
int mqR = 22000;
//rO sensor value
long rO = 41763;
//min value for Rs/Ro
float minRsRo = 0.358;
//max value for Rs/Ro
float maxRsRo = 2.428;
//sensor a coefficient value
float a = 116.6020682;
//sensor b coefficient value
float b = -2.769034857;

void setup() {
  pinMode(mqInput, INPUT);
  Serial.begin(9600);
}

void loop() {
  int adcRaw = analogRead(mqInput);
  long rS = ((1024.0 * mqR) / adcRaw) - mqR;
  Serial.print("Rs: ");
  Serial.println(rS);
  float rSrO = (float)rS / (float)rO;
  Serial.print("Rs/Ro: ");
  Serial.println(rSrO);
  if(rSrO < maxRsRo && rSrO > minRsRo) {
 float ppm = a * pow((float)rS / (float)rO, b);
 Serial.print("ppm: ");
 Serial.println(ppm);
  } else {
 Serial.println("Out of range.");
  }
  delay(1000);
}
```

Listing A.5: Código de Davide Girono para se utilizar com sensores MQ

```
# An R script to estimate MQ gas sensors correlation curve and compute Ro, min and max Rs/Ro


#DESCOMENTAR "#" PARA SE UTILIZAR DETERMINADAS SECCOES, COMO #POR EXEMPLO OS POINTSDATA QUE SAO RECOLHIDOS NO #WEBPLOTANALYZER



#
# Copyright (c) Davide Gironi, 2016
#
# Released under GPLv3.
# Please refer to LICENSE file for licensing information.

# How to use this script:
# 1) set limits as datasheet curve ("xlim" and "ylim")
#    ex.
#      xlim = c(10, 1000)
#      ylim = c(0.1, 10)
# 2) find out datasheet curve points, and write it out (to "pointsdata")
#    each line it's a point on cartesian coordinate system
#    the useful WebPlotDigitizer app can help you extract points from the graph
#    ex.
#      pointsdata = "
#        10.052112405371744, 2.283698378106183
#        20.171602728600178, 1.8052797165878915
#        30.099224396434586, 1.5715748803154423
#        50.09267987761949, 1.3195287228519417
#        80.38812026903305, 1.1281218760133969
#        90.12665922665023, 1.0815121769656304
#        100.52112405371739, 1.0430967861855598
#        199.62996638292853, 0.8000946404902397
#      "
# 3) optional for Ro estimation: measure the sensor resistance (set it to "mres" ohm value) at a know amount of gas
#    set it to 0 if you do not need the Ro estimation
#    ex.
#      mres = 26954
# 4) optional for Ro estimation: set the know amount of gas for the resistance measure of the previous step (to "mppm")
#    set it to 0 if you do not need the Ro estimation
#    ex.
#      mppm = 392
# 5) optional for min-max Rs/Ro estimation: set the minand max amount of gas the sensor will react to (as "minppm" and "maxppm")
#    set it to 0 if you do not need the min-max Rs/Ro estimation
#    ex.
#      minppm = 10
#      maxppm = 200
```

```
45 library(data.table)

47 #remove old variables
   rm(list=ls())
49
   #set input values
51 xlim = c(0.1, 10)
   ylim = c(0.1, 10)
53 minppm = 0
   maxppm = 0
55 mres = 0
   mppm = 0
57 pointsdata = "
   YOUR_POINTS_HERE
59 "

61 #load points using fread
   setnames(points <- fread(pointsdata, sep=",", sep2="\n"), c("x","y"))
63
   #set named list of points, and swapped list of points
65 #points will be used to plot and compute values as datasheet figure
   #pointsrev will be used to plot and compute values for the correlation function
       , it's the datasheet figure with swapped axis
67 x <- as.vector(points[,x])
   y <- as.vector(points[,y])
69 points = list(x=x, y=y)
   pointsrev = list(x=y, y=x)
71
   #the nls (Nonlinear Least Squares) it's used to perform the power regression on
       points
73 #in order to work, nls needs an estimation of staring values
   #we use log-log slope estimation to find intitial values
75
   #estimate fit curve initial values
77 xfirst = head(points$x, n=1)
   xlast = tail(points$x, n=1)
79 yfirst = head(points$y, n=1)
   ylast = tail(points$y, n=1)
81 bstart= log(ylast/yfirst)/log(xlast/xfirst)
   astart = yfirst/(xfirst^bstart)
83 #perform the fit
   fit <- nls("y~a*x^b", start=list(a=astart,b=bstart), data=points)
85
   #estimate fitref curve initial values
87 xfirstrev = head(pointsrev$x, n=1)
   xlastrev = tail(pointsrev$x, n=1)
89 yfirstrev = head(pointsrev$y, n=1)
   ylastrev = tail(pointsrev$y, n=1)
91 bstartrev = log(ylastrev/yfirstrev)/log(xlastrev/xfirstrev)
```

```
astartrev = yfirstrev/(xfirstrev^bstartrev)
fitrev <- nls("y~a*x^b", start=list(a=astartrev,b=bstartrev), data=pointsrev)

#plot fit curve (log-log scale)
fiteq = function(x){coef(fit)["a"]*x^(coef(fit)["b"])}
plot(points, log="xy", col="blue", xlab="ppm", ylab="Rs/Ro", xlim=xlim, ylim=
    ylim, panel.first=grid(equilogs=FALSE))
curve(fiteq, col="red", add=TRUE)

#plot fitrev curve (log-log scale)
fiteqrev = function(x){coef(fitrev)["a"]*x^(coef(fitrev)["b"])}
plot(pointsrev, log="xy", col="blue", xlab="Rs/Ro", ylab="ppm", xlim=ylim, ylim
    =xlim, panel.first=grid(equilogs=FALSE))
curve(fiteqrev, col="red", add=TRUE)

#plot fit curve (linear scale)
fiteq = function(x){coef(fit)["a"]*x^(coef(fit)["b"])}
plot(points, col="blue", xlab="ppm", ylab="Rs/Ro", panel.first=grid(equilogs=
    FALSE))
curve(fiteq, col="red", add=TRUE)

#plot fitrev curve (linear scale)
fiteqrev = function(x){coef(fitrev)["a"]*x^(coef(fitrev)["b"])}
plot(pointsrev, col="blue", xlab="Rs/Ro", ylab="ppm", panel.first=grid(equilogs
    =FALSE))
curve(fiteqrev, col="red", add=TRUE)

#estimate min Rs/Ro
cat("\nCorrelation function coefficients")
cat("\nEstimated a\n")
cat("  ")
cat(coef(fitrev)["a"])
cat("\nEstimated b\n")
cat("  ")
cat(coef(fitrev)["b"])
cat("\n")

#estimate min Rs/Ro
if (minppm != 0) {
    minRsRo = (maxppm/coef(fitrev)["a"])^(1/coef(fitrev)["b"])
    cat("\nEstimated min Rs/Ro\n")
    cat("  ")
    cat(minRsRo)
    cat("\n")
}

#estimate max Rs/Ro
if (maxppm != 0) {
    maxRsRo = (minppm/coef(fitrev)["a"])^(1/coef(fitrev)["b"])
```

```
137     cat("\nEstimated max Rs/Ro\n")
        cat("  ")
139     cat(maxRsRo)
        cat("\n")
141 }

143 #estimate Ro
    if (mppm != 0 && mres != 0) {
145     Ro = mres*(coef(fitrev)["a"]/mppm)^(1/coef(fitrev)["b"])
        cat("\nEstimated Ro\n")
147     cat("  ")
        cat(Ro)
149     cat("\n")
    }
```

Listing A.6: Código em R para se obter os coeficientes para o código de Davide Girono

Referências

[1] History Channel. The Great Smog of 1952 - HISTORY. URL: https://www.history.com/news/the-killer-fog-that-blanketed-london-60-years-ago.

[2] World Health Organization. WHO Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide: global update 2005: summary of risk assessment. *Geneva: World Health Organization*, 2006. arXiv:arXiv:1011.1669v3, doi:10.1016/0004-6981(88)90109-6.

[3] Michael Guarnieri e John R. Balmes. Outdoor air pollution and asthma, 2014. arXiv:arXiv:1011.1669v3, doi:10.1016/S0140-6736(14)60617-6.

[4] Dana Loomis, Yann Grosse, Béatrice Lauby-Secretan, Fatiha El Ghissassi, Véronique Bouvard, Lamia Benbrahim-Tallaa, Neela Guha, Robert Baan, Heidi Mattock, e Kurt Straif. The carcinogenicity of outdoor air pollution. *The Lancet Oncology*, 2013. arXiv:S1470-2045(13)70487-X, doi:10.1016/S1470-2045(13)70487-X.

[5] T Rotko, L Oglesby, N Kunzli, e M J Jantunen. Population sampling in European air pollution exposure study, EXPOLIS: comparisons between the cities and representativeness of the samples. *Journal of Exposure Analysis and Environmental Epidemiology*, 2000. doi:DOI 10.1038/sj.jea.7500101.

[6] M. I. Mead, O. a M Popoola, G. B. Stewart, P. Landshoff, M. Calleja, M. Hayes, J. J. Baldovi, M. W. McLeod, T. F. Hodgson, J. Dicks, a. Lewis, J. Cohen, R. Baron, J. R. Saffell, e R. L. Jones. The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks. *Atmospheric Environment*, 2013. doi:10.1016/j.atmosenv.2012.11.060.

[7] Nihal Kularatna e B. H. Sudantha. An Environmental Air Pollution Monitoring System Based on the IEEE 1451 Standard for Low Cost Requirements. *IEEE Sensors Journal*, 2008. doi:10.1109/JSEN.2008.917477.

[8] K. Grahammer, M.D. Jawson, e J. Skopp. Day and night soil respiration from a grassland. *Soil Biology and Biochemistry*, 23(1):77–81, jan 1991. URL: https://www.sciencedirect.com/science/article/pii/003807179190165G?via{%}3Dihub, doi:10.1016/0038-0717(91)90165-G.

[9] Sherin Abraham e Xinrong Li. A cost-effective wireless sensor network system for indoor air quality monitoring applications. Em *Procedia Computer Science*, 2014. doi:10.1016/j.procs.2014.07.090.

[10] R. Piedrahita, Y. Xiang, N. Masson, J. Ortega, A. Collier, Y. Jiang, K. Li, R. P. Dick, Q. Lv, M. Hannigan, e L. Shang. The next generation of low-cost personal air quality sensors for quantitative exposure monitoring. *Atmospheric Measurement Techniques*, 2014. doi:10.5194/amt-7-3325-2014.

[11] Earth System Research Laboratory US Department of Commerce, NOAA. ESRL Global Monitoring Division - Global Greenhouse Gas Reference Network. URL: https://www.esrl.noaa.gov/gmd/ccgg/trends/.

[12] Health and Safety Executive. EH40/2005 Workplace exposure Limits. Relatório técnico, 2011. URL: http://www.hse.gov.uk/pubns/priced/eh40.pdf.

[13] O. A. Seppanen, W. J. Fisk, e M. J. Mendell. Association of Ventilation Rates and CO2 Concentrations with Health andOther Responses in Commercial and Institutional Buildings. *Indoor Air*, 9(4):226–252, dec 1999. URL: http://doi.wiley.com/10.1111/j.1600-0668.1999.00003.x, doi:10.1111/j.1600-0668.1999.00003.x.

[14] A. Chevalier, F. Gheusi, R. Delmas, C. Ordóñez, C. Sarrat, R. Zbinden, V. Thouret, G. Athier, e J. M. Cousin. Influence of altitude on ozone levels and variability in the lower troposphere: A ground-based study for western Europe over the period 2001-2002. *Atmospheric Chemistry and Physics*, 2007. doi:10.5194/acp-7-4311-2007.

[15] Claire E. Reeves, Stuart A. Penkett, Stephane Bauguitte, Kathy S. Law, Mathew J. Evans, Brian J. Bandy, Paul S. Monks, Gavin D. Edwards, Gavin Phillips, Hannah Barjat, Joss Kent, Ken Dewey, Sandra Schmitgen, e Dieter Kley. Potential for photochemical ozone formation in the troposphere over the North Atlantic as derived from aircraft observations during ACSOE. *Journal of Geophysical Research D: Atmospheres*, 2002. doi:10.1029/2002JD002415.

[16] R J Delfino, A M Murphy-Moulton, e M R Becklake. Emergency room visits for respiratory illnesses among the elderly in Montreal: association with low level ozone exposure. *Environmental research*, 1998. doi:10.1006/enrs.1997.3794.

[17] Joseph R. Stetter e Jing Li. Amperometric gas sensors - A review, 2008. doi:10.1021/cr0681039.

[18] Irja Helm, Lauri Jalukse, e Ivo Leito. Measurement uncertainty estimation in amperometric sensors: A tutorial review, 2010. doi:10.3390/s100504430.

[19] S. C. Chang, J. R. Stetter, e C. S. Cha. Amperometric gas sensors, 1993. doi:10.1016/0039-9140(93)80002-9.

[20] Jose María, Muñoz Martín, María Del Mar Baeza, e Francisco Céspedes. Advanced amperometric nanocomposite sensors based on carbon nanotubes and graphene: Characterization, Optimization, Functionalization and Applications Thesis to opt for the PhD in Chemistry. Relatório técnico, 2015. URL: https://www.tesisenred.net/bitstream/handle/10803/311424/jmmm1de1.pdf?sequence=1{&}isAllowed=y.

[21] Chengxiang Wang, Longwei Yin, Luyuan Zhang, Dong Xiang, e Rui Gao. Metal oxide gas sensors: Sensitivity and influencing factors, 2010. doi:10.3390/s100302088.

[22] George F. Fine, Leon M. Cavanagh, Ayo Afonja, e Russell Binions. Metal oxide semiconductor gas sensors in environmental monitoring, 2010. doi:10.3390/s100605469.

[23] Xiao Liu, Sitian Cheng, Hong Liu, Sha Hu, Daqiang Zhang, e Huansheng Ning. A Survey on Gas Sensing Technology. *Sensors*, 2012. arXiv:1305.7427, doi:10.3390/s120709635.

[24] SeungHwan Yi, JongSeon Park, e JeongMin Park. Temperature compensation of novel NDIR CO2 gas sensor. Em *Proceedings of IEEE Sensors*, 2006. doi:10.1109/ICSENS.2007.355886.

[25] Trieu Vuong Dinh, In Young Choi, Youn Suk Son, e Jo Chun Kim. A review on non-dispersive infrared gas sensors: Improvement of sensor detection limit and interference correction, 2016. doi:10.1016/j.snb.2016.03.040.

[26] Dejan Nedelkovski. DHT11 & DHT22 Sensor Temperature and Humidity Tutorial, 2016. URL: http://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/.

[27] ETSI. ETSI - GPRS. URL: https://www.etsi.org/technologies-clusters/technologies/mobile/gprs?highlight=YToyOntpOjA7czo0OiJncHJzIjtpOjE7czo2OiJncHJzJ3MiO30=.

[28] ETSI. ETSI - GSM. URL: https://www.etsi.org/technologies-clusters/technologies/mobile/gsm.

[29] Stephen Feinberg. CAIRSENSE Study: Real-world evaluation of low cost sensors in Denver, Colorado. Relatório técnico, 2016. URL: https://www.epa.gov/sites/production/files/2016-10/documents/real{_}world{_}evaluation.pdf.

[30] DC cable. DC Cable Sizing Tool - Wire Size Calculator - MM2 & AWG - solar-wind.co.uk. URL: http://www.solar-wind.co.uk/cable-sizing-DC-cables.html.

[31] DEVICE PLUS. Arduino Communication Protocols Tutorial. URL: https://www.deviceplus.com/how-tos/arduino-guide/arduino-communication-protocols-tutorial/.

[32] Winsen. MG811 Solid Electrolyte CO2 Gas Sensor. URL: https://www.compel.ru/pdf-items/winsen/pn/mg811-solid-electrolyte-co2-gas-sensor/b813a7da729d069bcb7b9b7b425b16b3.

[33] Winsen. Ozone Gas Sensor ModelMQ131 Low Concentration Manual. Relatório técnico, 2014. URL: www.winsen-sensor.comhttps://www.icbanq.com/data/ICBShop/board/MQ131OzoneGasSensor(LowConcentration).pdf.

[34] Arduino. Arduino Playground - MQGasSensors. URL: https://playground.arduino.cc/Main/MQGasSensors.

[35] Milan Romic. Problem PH-NET SIM – Flashing SIM900A to SIM900 « Easy for Engineers. URL: http://www.easy4engineers.com/problem-ph-net-sim-flashing-sim900a-to-sim900/.

[36] Andy. SIM900A fixed for Europe | amichalec.net :: homepage. URL: http://amichalec.net/2014/08/sim900a-fixed-for-europe/.

[37] SIM Tech. SIMCom Wireless Solutions. URL: http://simcomm2m.com/En/.

[38] Onsetcomp. Telaire 7001 Manufacturers Manual | Onset Data Loggers. URL: https://www.onsetcomp.com/support/tech-note/telaire-7001-manufacturers-manual.

[39] Aeroqual. Foreword. Relatório técnico, 2004. URL: www.aeroqual.com.

[40] Yair Altman. export_fig - File Exchange - MATLAB Central. URL: https://www.mathworks.com/matlabcentral/fileexchange/23629-export{_}fig.

[41] AIQCN. Poluição do ar em Portugal: Mapa da qualidade do ar em tempo real. URL: http://aqicn.org/map/portugal/pt/{#}@g/41.8159/-7.6904/6z.

[42] Ankit Rohatgi. WebPlotDigitizer - extract data from plots, images, and maps, 2010. URL: http://arohatgi.info/WebPlotDigitizer/.

[43] Davide Gironi. Davide Gironi: MQ gas sensor correlation function estimation by datasheet. URL: http://davidegironi.blogspot.com/2017/05/mq-gas-sensor-correlation-function.html{#}.W6QLJWhKiiM.

[44] SandBox. MG-811 CO2 Sensor Module | Sandbox Electronics. URL: http://sandboxelectronics.com/?p=147.

[45] Arduíno Forums. 6v to DC input jack seems to work fine? URL: https://forum.arduino.cc/index.php?topic=124834.0.