

Implementação de um Array de Detetores de Radiação Ionizante

Ana Cláudia da Silva Oliveira

TESE DE MESTRADO

Tese submetida ao Instituto Superior de Engenharia do Porto
para a obtenção do grau de Mestre em
Engenharia de Computação e Instrumentação Médica

Orientadores

DOUTOR CRISTIANO ABREU

Professor Adjunto do Departamento de Física
Instituto Superior de Engenharia do Porto

DOUTOR CARLOS VINHAIS

Professor Adjunto do Departamento de Física
Instituto Superior de Engenharia do Porto

Porto, 18 de Outubro de 2013

*"O único lugar onde o sucesso vem antes
do trabalho é no dicionário"*
Albert Einstein

Agradecimentos

Agradeço ao Doutor Cristiano Abreu pela possibilidade que me ofereceu de trabalhar nesta área, que superou sem qualquer dúvida as minhas expectativas. Agradeço também pelos ensinamentos, pela motivação e disponibilidade, que garantiram todo o trabalho realizado.

Agradeço ao Doutor Carlos Vinhais por toda a disponibilidade, acompanhamento e atenção que dedicou durante o desenvolvimento deste trabalho. Agradeço toda a paciência, colaboração e dedicação, que sempre me motivaram para a concretização do objetivo final.

Um agradecimento muito especial aos meus pais por todo apoio que me deram ao longo de toda a minha vida e pela confiança que depositaram no meu percurso académico. Agradeço por permitirem que percorresse este caminho e por estarem ao meu lado em todos os momentos.

Agradeço ao meu irmão Jorge pela disponibilidade em qualquer momento, pela ajuda e apoio. Por me ensinar a resolver problemas da forma mais fácil e a manter a calma nos momentos mais difíceis.

À minha família e aos meus amigos por sempre acreditarem em mim e pela presença na minha vida, garantindo-me forças para continuar o meu percurso.

Resumo

Atualmente, as radiações ionizantes desempenham um papel fundamental nas áreas de diagnóstico e terapia, estando omnipresentes em ambientes hospitalares. Contudo, devido aos efeitos biológicos adversos da radiação, torna-se essencial a proteção dos profissionais de saúde e pacientes. Consequentemente, um *array* de detetores capazes de produzir um sinal acústico, aquando da presença de radiação ionizante excedendo determinados valores limite e transmissão via *wireless* das leituras para um sistema central é de grande interesse prático.

Nesta dissertação, foi implementado um sistema capaz de alimentar um *array* de sensores de radiação para monitorização de diferentes espaços e transmissão das leituras efetuadas via *wireless*. A aquisição de dados foi realizada, recorrendo à utilização de um conversor analógico-digital. Vários testes de validação foram realizados, através de vários passos para alcançar a concretização do sistema final, nomeadamente testes relativos ao circuito de deteção, módulos de comunicação *wireless*, bem como o uso de diferentes ambientes de desenvolvimento integrados (IDE). Os resultados destes testes mostram a visualização e gravação adequadas dos dados relativos aos níveis de radiação, bem como a transmissão de dados de forma viável, permitindo a monitorização de espaços sujeitos à presença de radiação ionizante.

Desta forma, um *array* de contadores Geiger-Müller, ligados a módulos *wireless* XBee *open-source* e uma placa Arduino, possibilitou a implementação de um sistema viável e de baixo custo para monitorização de radiação ionizante e registar esses mesmos dados para posterior análise.

Palavras-chave: Radiação Ionizante; Geiger-Müller; Monitorização; Arduino; ZigBee.

Abstract

Ionizing radiations are relevant today due to their application in diagnosis and therapy and, therefore, are omnipresent in hospitals' environments. Nonetheless, due to the adverse biological effects of radiation, there is a need for protection of health care professionals and patients. Accordingly, an array of detectors able to produce an audible signal in the presence of radiation, exceeding threshold levels, and to transmit wirelessly the readings to a centralized system is of great practical interests.

In this thesis a system capable of powering an array of radiation sensors to monitor different spaces and communicate the readings wirelessly has been implemented. The data acquisition was carried out by using an analog-to-digital converter. Several validation tests were carried out throughout the many steps leading to the final system, namely concerning the detection circuit, wireless communication modules, as well as the use of different integrated development environments (IDE). Results from these tests were aimed toward an adequate visualization and data recording of the radiation levels, monitored from different spaces.

Thus, an array of Geiger-Müller counters, linked together by open-source XBee wireless modules and Arduino boards, made possible to implement a cost-effective and viable system to detect the presence of ionizing radiation and register it for latter analysis.

Key-words: Ionizing Radiation; Geiger-Müller; Monitoring; Arduino; ZigBee.

Conteúdo

Agradecimentos	vi
Resumo	viii
Abstract	x
Conteúdo	xiii
Lista de Figuras	xvii
Lista de Tabelas	xix
Abreviaturas	xxi
1. Introdução	1
1.1 Motivação e Enquadramento	1
1.2 Objetivo	2
1.3 Estrutura	2
2. Radioatividade	5
2.1 Decaimento radioativo	6
2.1.1 Número de massa excessivo	6
2.1.2 Número elevado de neutrões	7
2.1.3 Número elevado de prótons	8
2.1.4 Excesso de Energia	10
2.2 Interação da Radiação com a Matéria	12
2.2.1 Interação dos fótons com a matéria	12
2.2.2 Interação de Partículas Carregadas com a matéria	15
3. Radioproteção	17
3.1 Efeitos Biológicos da Radiação	17
3.1.1 Classificação	18
3.1.2 Modos de Ação	18

3.1.3	Tipos de Efeitos	19
3.2	Detetores de fase gasosa	20
3.2.1	Regiões de Operação dos Detetores Gasosos	21
3.2.2	Câmara de Ionização	23
3.2.3	Contador Proporcional	25
3.3	Detetor de Geiger Muller	25
3.3.1	Estrutura	26
3.3.2	Princípios de Funcionamento	26
4.	Implementação do sistema	29
4.1	Desenho do Sistema	29
4.2	Circuito Eletrônico	30
4.2.1	Fonte de Alimentação	34
4.2.2	Alimentação do Tubo GM	35
4.2.3	Produção do sinal analógico para aquisição e geração de um sinal acústico	38
4.2.4	Circuito Impresso PCB	41
4.3	Aquisição de Dados	42
4.3.1	Conversor Analógico-Digital	43
4.4	Comunicação de dados	46
4.4.1	Tipos de Comunicação	46
4.4.2	Transmissão de Dados	46
4.4.3	Redes Sem-fios	47
5.	Aquisição de Dados	51
5.1	Especificações do Arduino	51
5.2	Ambientes de Desenvolvimento Integrado	55
5.2.1	Arduino	57
5.2.2	Processing	57
5.2.3	Python	57
5.3	Testes de Funcionalidade	58
5.3.1	Teste 1	58
5.3.2	Teste 2	60
5.3.3	Teste 3	62
6.	Redes Sem-Fios	65
6.1	ZigBee	65
6.1.1	Modelo de Camadas	66
6.1.2	Tipos de Nós	69
6.1.3	Dispositivos numa rede <i>ZigBee</i>	69
6.1.4	Topologias de Rede	70
6.2	Módulos XBee	72
6.2.1	Pinos dos Módulos XBee	73
6.2.2	Shield XBee	73
6.2.3	Antenas	75

6.2.4	Adaptadores	76
6.3	Configurações dos Módulos XBee	80
6.3.1	Modos de Operação	80
6.3.2	Configuração no X-CTU	84
6.4	Testes de Funcionalidade	85
6.4.1	Teste 1	86
6.4.2	Teste 2	88
6.4.3	Teste 3	90
6.4.4	Teste 4	92
6.4.5	Teste 5	94
6.4.6	Teste 6	94
6.4.7	Teste 7	95
6.4.8	Teste 8	98
6.4.9	Teste 9	98
7.	Conclusão	103
	Bibliografia	110
A.	Aquisição: Código Teste 2	111
B.	Código Ascii	115
C.	Guia de Informações XBee	117

Lista de Figuras

2.1	Fissão Nuclear	6
2.2	Decaimento Alfa	7
2.3	Decaimento Beta	8
2.4	Captura Eletrónica	9
2.5	Emissão Gama	10
2.6	Conversão Interna	11
2.7	Gráfico representativo da interação dos fótons com a matéria	12
2.8	Espalhamento de <i>Compton</i>	13
2.9	Efeito Fotoelétrico	14
2.10	Produção de Pares	14
2.11	Atenuação dos fótons na matéria	15
3.1	Ação da radiação ionizante na estrutura da molécula de ADN	19
3.2	Detetor Gasoso	21
3.3	Regiões de operação dos detetores gasosos	22
3.4	Fenómeno de amplificação do gás de deteção	22
3.5	Tipos de câmaras de ionização	24
3.6	Contador de <i>Geiger-Muller</i>	25
3.7	Representação esquemática do fenómeno de avalanches de <i>Townsend</i>	27
3.8	Função do gás de <i>quenching</i> no interior do tubo	27
4.1	Diagrama de Blocos	30
4.2	Esquema do circuito eletrónico implementado	32
4.3	Circuito eletrónico implementado	33
4.4	Alimentação do circuito de deteção	34
4.5	Circuito de alimentação do tubo GM	35
4.6	Circuito integrado 555	36
4.7	Sinal de saída do pino 3 do CI 555	36
4.8	Ligações ao tubo GM	38
4.9	Circuito responsável pela produção do sinal analógico para aquisição e geração de um sinal acústico	40
4.10	Sinal de saída do pino 1 do <i>Schmitt Trigger</i>	41
4.11	Placa de prototipagem com os componentes soldados	42

4.12	Representação esquemática de um conversor analógico-digital (ADC)	43
4.13	Conversores analógicos-digitais utilizados	45
4.14	Representação simplificada da comunicação em série entre o Arduino e o PC	47
5.1	Modelos Arduino utilizados	52
5.2	Identificação dos constituintes do Arduino	54
5.3	Descrição dos pinos do microcontrolador ATmega328	55
5.4	Ambientes de desenvolvimento integrado (IDE) utilizados	56
5.5	<i>Serial Monitor</i> na fase de testes onde estão representados alguns dos dados provenientes do circuito eletrónico	59
5.6	Resultados do teste 2	61
5.7	Resultados do teste 3	64
6.1	Camadas do padrão ZigBee	66
6.2	Topologia <i>Per</i>	70
6.3	Topologia <i>Star</i>	71
6.4	Topologia <i>Mesh</i>	71
6.5	Topologia <i>Cluster Tree</i>	72
6.6	Pinos do módulo XBee	73
6.7	<i>Shield Xbee Wireless SD</i>	75
6.8	Tipos de antenas dos módulos XBee	76
6.9	Adaptador com o módulo XBee e cabo FTDI conectados	77
6.10	Procedimento para a soldadura dos componentes	79
6.11	Adaptador com os componentes soldados	80
6.12	Pinos do adaptador para ligação ao cabo FTDI	80
6.13	Estrutura geral de uma <i>frame</i> API	81
6.14	<i>Frame</i> para o pacote I/O RX	82
6.15	Ilustração das ligações entre o Arduino e módulo XBee, para configuração do módulo com recurso ao <i>software</i> X-CTU	85
6.16	Desenho das conexões para o teste 1 (<i>Chat</i>)	86
6.17	Configuração X-CTU e resultados do teste 1 (<i>"Chat"</i>)	87
6.18	Definições do teste 2	89
6.19	Desenho e implementação do teste 2	90
6.20	Resultados do teste 3	91
6.21	Resultados do teste 4	93
6.22	Implementação do teste 4	94
6.23	Implementação e resultados do teste 6	95
6.24	Implementação do teste 7	96
6.25	Resultados do teste 7	97
6.26	Implementação e resultados do teste 8	98
6.27	Representação esquemática da implementação final do sistema de detetores de radiação ionizante	99
6.28	Implementação do sistema de detetores de radiação ionizante	101

6.29 Interface gráfica simples implementada em Processing, para monitorização das contagens do <i>array</i> de detetores	102
B.1 Código Ascii	116

Lista de Tabelas

2.1	Propriedades das partículas radioativas	11
4.1	Componentes Eletrônicos para implementação do circuito de detecção	31
4.2	Descrição dos pinos do CI 555	37
4.3	Comparação entre os protocolos de comunicação <i>Wi-Fi</i> , <i>Bluetooth</i> e <i>ZigBee</i>	49
5.1	Características dos modelos Arduino Duemilanove e Uno	52
6.1	Funções dos ZC, ZR e ZED na camada de rede.	69
6.2	Descrição dos pinos do módulo XBee	74
6.3	Lista de Componentes que compõem o adaptador XBee da Adafruit	78
6.4	Especificações para o <i>frame</i> I/O RX	83
6.5	Parâmetros a definir para a porta COM	84

Lista de Abreviaturas

A	Número de Massa/Massa Atómica
AC	Corrente Alternada
ADC	<i>Analog Digital Converter</i>
ADN	Ácido desoxirribonucleico
AES	<i>Advanced Encryption Standard</i>
API	<i>Application Programming Interface</i>
APOs	<i>Application Objects</i>
AT	<i>Transparent Mode</i>
CHL	<i>Cluster Head</i>
CID	<i>Cluster Identifier</i>
CSMA–CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CTS	<i>Clear to Send</i>
DC	Corrente Contínua
DSS	<i>Direct Sequence Spread Spectrum</i>
FFD	<i>Full-function device</i>
GM	<i>Geiger-Müller</i>
GND	<i>Ground</i>
I/O	<i>Input</i>
IC	<i>Integrated Circuit</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
LDR	<i>Light Dependent Resistor</i>
LET	<i>Linear Energy Transfer</i>
LSB	<i>Least Significant Byte</i>
MAC	<i>Medium Access Control</i>
MSB	<i>Most Significant Byte</i>
NWK	<i>Network</i>
OSI/ISO	<i>International Organization for Standardization</i>

PAN	<i>Personal Area Network</i>
PC	<i>Personal Computer</i>
PCB	<i>Printed Circuit Board</i>
PHY	<i>Physical</i>
PWM	<i>Pulse Width Modulation</i>
RF	Rádio Frequência
RFD	<i>Reduced-Function Devices</i>
RSSF	Rede de Sensores sem Fios
RSSI	<i>Received signal strength indication</i>
RX	<i>Receiver</i>
TX	<i>Transmitter</i>
USART	<i>Universal Synchronous Asynchronous Receiver Transmitter</i>
Vcc	Fonte de Alimentação
WLAN	<i>Wireless Local Area Network</i>
WMAN	<i>Wireless Metropolitan Area Network</i>
WPAN	<i>Wireless Personal Area Network</i>
WWAN	<i>Wireless Wide Area Network</i>
ZC	<i>ZigBee Coordinator</i>
ZDO	<i>ZigBee Device Object</i>
ZED	<i>ZigBee End Device</i>
ZR	<i>ZigBee Router</i>

Introdução

1.1 Motivação e Enquadramento

Ao longo dos anos o ser humano tem estado constantemente em contacto com a radiação proveniente do meio ambiente. Devido a uma evolução tecnológica colossal, as situações de exposição à radiação aumentaram, não só devido ao ambiente, mas também devido a situações mais específicas, tais como estações nucleares e ambientes hospitalares com equipamentos de cariz radiológico. A utilização da radiação nos equipamentos hospitalares é considerada benéfica, devido às suas características terapêuticas e de diagnóstico. No entanto, os efeitos da radiação no corpo humano são bastante complexos, dependendo da natureza e intensidade da radiação. Com o intuito de evitar este tipo de situações, foi necessária a criação de medidas capazes de minimizar ou mesmo evitar eventuais situações desta índole. Como é sobejamente conhecido, a radiação é um fenómeno não identificado pelos sentidos humanos e, por esse motivo, as medidas preventivas e de proteção deverão ser permanentes, isto é, deve existir uma monitorização do espaço físico onde poderá existir radiação ionizante, bem como a proteção constante de trabalhadores ligados à área e pacientes, no caso de unidades hospitalares.

A necessidade crescente da prevenção e proteção contra a radiação ionizante levou à criação de medidas e normas que asseguram a integridade física das pessoas expostas. Face a esta necessidade, surgiu a ideia de monitorização de espaços onde a presença de radiação ionizante é suficiente para provocar danos biológicos aos seres humanos. Os espaços físicos sujeitos à presença de radiação ionizante requerem a implementação de sistemas de controlo eficazes, credíveis, específicos, duradouros e

de qualidade. A utilização de sensores de elevada sensibilidade, de pequeno porte e baixo custo são uma mais-valia nestes sistemas de controlo, permitindo a monitorização dos espaços físicos. Neste âmbito, as redes sem-fios são consideradas uma solução *Gold Standard*, oferecendo recursos válidos e eficientes para controlo. O conceito de Redes de Sensores sem Fios (RSSF) ou *Wireless* é então introduzido, apresentando-se como uma tecnologia emergente que oferece aos seus utilizadores a possibilidade de monitorizar diferentes espaços físicos, através de um único dispositivo local. Desta forma, é possível o desenvolvimento de um *array* de detetores de radiação ionizante, constituindo uma rede de sensores conectados via *wireless*, tornando-se uma solução pertinente no que diz respeito à Radioproteção.

1.2 Objetivo

O presente trabalho consiste na integração de tecnologias para desenvolvimento de uma rede de sensores sem-fios, com o intuito de monitorizar um espaço físico sujeito à presença de radiação ionizante.

Para garantir a implementação de um sistema deste tipo, os objetivos específicos deste trabalho são: construir um circuito electrónico que assegure não só a alimentação de um sensor de radiação ionizante como também a produção de um sinal sonoro para alarme; desenvolver soluções para aquisição de dados, para garantir a monitorização constante dos espaços onde os sensores são implementados; e desenvolver mecanismos de controlo sem-fios, através de dispositivos que permitem a comunicação via *wireless*.

1.3 Estrutura

Este trabalho está dividido em 7 Capítulos. O capítulo 2 e 3 constituem uma abordagem teórica ao tema, incluindo dois conceitos essenciais neste trabalho: Radioatividade e Radioproteção. O Capítulo 4 corresponde à apresentação do sistema desenvolvido. O capítulo 5 prende-se com a informação e resultados associados à aquisição de dados. O capítulo 6 correspondente a todo o processo necessário para a criação de um *array* de sensores de radiação ionizante. Por fim, o capítulo 7, onde

são apresentadas as conclusões do trabalho realizado.

No Capítulo 2, insere-se um estudo referente ao fenómeno da Radioatividade e a sua contextualização. Este é dividido em duas Secções, 2.1 e 2.2, que consistem na descrição do decaimento radioativo e da interação da radiação com a matéria, onde são apresentados vários tipos de radiações ionizantes, bem como o resultado da sua interação com a matéria.

O capítulo 3 surge com o intuito de apresentar soluções na deteção da radiação ionizante. Foi dividido em quatro Secções: efeitos biológicos da radiação, detetores gasosos e detetor de *Geiger-Müller*. A primeira secção 3.1, expõe os efeitos da radiação nos tecidos humanos, que podem ser classificados formas diferentes, modos de ação distintos e tipos de efeitos. A secção 3.2 corresponde à descrição dos detetores gasosos e suas regiões de operação. A secção 3.3 incide na descrição e funcionamento do detetor *Geiger-Müller*.

O capítulo 4 descreve a solução implementada, sendo dividido em quatro secções: desenho do sistema, circuito eletrónico, aquisição e comunicação. A secção 4.1, de uma forma geral, introduz a funcionalidade do sistema e a sua organização. A secção 4.2 descreve a forma como o circuito eletrónico foi implementado. A secção acerca da aquisição, 4.3, aborda o conceito de conversor analógico-digital. A última secção 4.4 deste Capítulo, aborda sucintamente as várias possibilidades de comunicação, para transmissão de dados.

O capítulo 5 está relacionado com todo o processo que assegura a aquisição de dados do sistema e está repartido em especificações do Arduino, ambientes de desenvolvimento integrado e testes de funcionalidade. Na secção 5.1 está presente a análise das características gerais dos modelos do Arduino utilizados. Na secção 5.2 são brevemente explicados os ambientes de desenvolvimento integrado (IDE) utilizados para aquisição, leitura e amostragem de dados. Na secção 5.3 são apresentados alguns dos resultados de testes realizados para estas componentes do sistema.

O capítulo 6 engloba três importantes secções: *ZigBee*, módulo XBee e configurações. Na primeira secção deste capítulo, secção 6.1, são apresentadas as redes *ZigBee*. Na secção 6.2, correspondente ao módulo XBee, são descritas as especificações para este dispositivo. Para finalizar, na secção 6.3, são apresentadas as

configurações realizadas para a construção das redes e alguns dos testes realizados nesta fase.

No último capítulo, correspondente ao capítulo 7 é feita uma conclusão de todo o trabalho realizado, bem como uma breve exposição do trabalho futuro.

Radioatividade

O presente capítulo está dividido em duas partes que se inserem no âmbito da radioatividade: decaimento radioativo (secção 2.1) e interação da radiação com a matéria (secção 2.2). Como introdução ao tema, é apresentada uma resenha histórica, para posteriormente serem abordadas as questões relativas aos dois temas aqui descritos.

O fenómeno da radioatividade foi descoberto por Henri Becquerel em 1896, criando uma nova porta no que diz respeito a esta área, conseguindo oportunidades para o desenvolvimento de novas experiências [1]. A radioatividade é um fenómeno natural ou artificial, em que substâncias ditas radioativas possuem a capacidade de emissão de radiação. Isto é, eventualmente a parte central do átomo, o núcleo, possui uma energia considerada elevada e não sustentável para o átomo. Este núcleo instável torna o átomo radioativo. Para conseguir libertar esta quantidade de energia e tornar-se estável, o átomo realiza o processo de emissão de radiação, assim o núcleo com Z prótons e N neutrões transformar-se-á num núcleo com novos valores de Z e N , transformando-se num nuclídeo diferente [2]. A descoberta da radioatividade artificial iniciou uma nova linha de pesquisas nucleares. Desde aí, foi produzida uma enorme quantidade de nuclídeos provenientes de várias reações nucleares, cerca de 2700 radionuclídeos produzidos artificialmente em ciclotrões e reatores nucleares. A radioatividade artificial foi apenas produzida a partir de 1934. Os primórdios da produção de radioatividade artificial deve-se ao trabalho de Irène Curie e Frédéric Joliot que tornaram alumínio, boro e magnésio radioativos, através do seu bombardeamento com partículas alfa de polónio. Na Terra, um dos mais importantes nuclídeos naturais do grande grupo existente é o potássio, com uma

abundância isotópica de cerca de 1,0118% e um tempo de meia vida de $1,22 \times 10^9$ anos. Todos os elementos com número atômico (Z) superior a 83 são radioativos [2].

2.1 Decaimento radioativo

Um núcleo que não é estável, como já foi referido, irá libertar-se da energia em excesso, através da ejeção de partes do núcleo por conversão interna ou por emissão de energia sob a forma de fótons (raios gama, γ), a este processo dá-se o nome de decaimento radioativo. A sua estabilidade depende da sua organização interna e também da energia de ligação entre os nucleões [3].

O tipo de decaimento radioativo depende de vários aspetos, tais como o excesso de massa nuclear (fissão e decaimento alfa), taxa de prótons elevada (decaimento beta), taxa de neutrões elevada (decaimento por positrão ou beta+ e captura Eletrónica) e excesso de energia (emissão gama e conversão interna) [4].

2.1.1 Número de massa excessivo

Fissão

Este é um processo destrutivo em que núcleos pesados instáveis se partem em dois fragmentos, acompanhados pela emissão de dois ou três neutrões e calor (Figura 2.1). Os neutrões são responsáveis por transportar uma energia de cerca de 1,5 MeV. O processo de fissão pode ocorrer não só de forma espontânea como também pelo bombardeamento com partículas de alta energia [5].

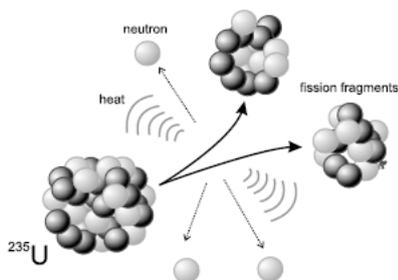


Fig. 2.1: Fissão nuclear [4].

Decaimento alfa (α)

Em 1899 foi descoberto e investigado o primeiro modo de decaimento, o decaimento α . Este ocorre mais frequentemente em núclídeos pesados, tais como o urânio, radão, plutônio, entre outros e consiste na emissão de uma partícula alfa (Figura 2.2). Os átomos que sofrem este tipo de decaimento possuem uma massa atômica (A) elevada. Após a emissão da partícula α , específica deste tipo de decaimento, o número atômico (Z) e número de massa (A) relativamente ao radionuclídeo pai, diminui respectivamente duas e quatro unidades. Portanto, a partícula alfa possui uma carga positiva (+2) e quatro unidades de massa atômica (${}^4_2\text{He}$). A Eq. 2.1 representa um exemplo de decaimento α [3]:

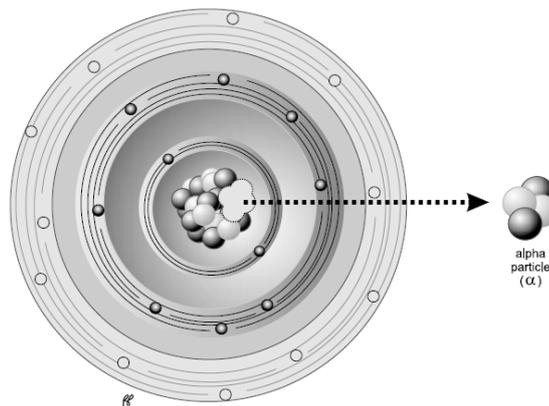
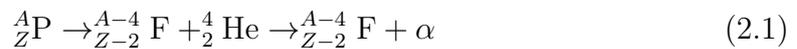


Fig. 2.2: Decaimento alfa [4].

2.1.2 Número elevado de neutrões

Decaimento beta menos (β^-)

Quando um radionuclídeo é rico em neutrões ocorre o decaimento do tipo β^- . Este processo consiste na conversão de um neutrão num próton e emissão de uma partícula β^- e de um antineutrino $\bar{\nu}$. A Figura 2.3 (a) representa esquematicamente o processo de decaimento β^- [3, 6]. A partícula β^- é similar a um eletrão, difere deste no sentido em que provém do núcleo do átomo, contrariamente ao eletrão que provém das suas orbitais [2]. Assim, o átomo-filho fica com mais um próton que o

átomo-pai. Este tipo de decaimento pode ser descrito através da Eq. 2.2 [5]:



Referir o motivo que conduz à emissão de um antineutrino é também essencial. Assim sendo, a diferença de energia entre os núclídeos pai e filho é designada energia de decaimento (energia máxima). A média de energia cinética das partículas β^- é cerca de 1/3 da energia de decaimento, o que prova que as partículas β^- transportam frequentemente apenas parte da energia de decaimento. Para satisfazer a lei de conservação da energia, foi postulada uma nova partícula, o antineutrino, sem carga e massa desprezável, que deve transportar a restante energia em cada decaimento β^- [2, 1].

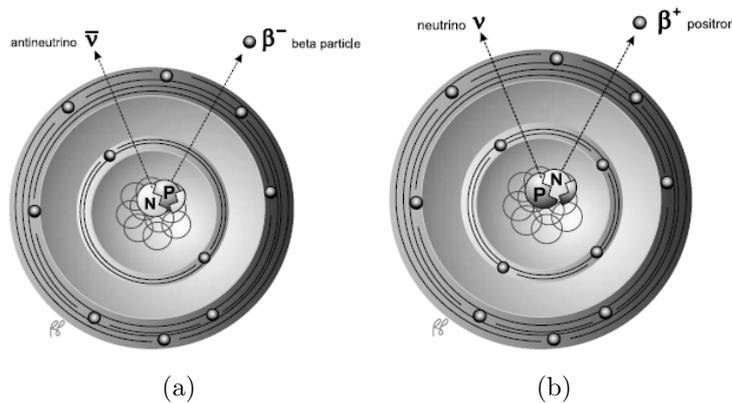


Fig. 2.3: Decaimento Beta: (a) Decaimento β^- : emissão de uma partícula β^- e de um antineutrino; (b) Decaimento β^+ : emissão de uma partícula β^+ e de um neutrino [4].

2.1.3 Número elevado de prótons

Decaimento beta mais (β^+)

Quando um radionuclídeo é rico em prótons, ocorre um decaimento por emissão de um positrão (similar ao electrão mas com carga positiva, ou seja, é a sua antipartícula) e um neutrino, ν (antipartícula do antineutrino), como é possível verificar através da observação da Figura 2.3 (b). A emissão de positrão acontece apenas quando a energia de decaimento entre os núclídeos pai e filho é superior a

1.02 MeV. Contrariamente ao decaimento β^- , um próton é convertido em neutrão, sendo emitido um positrão, designado neste caso como partícula β^+ e, portanto, o número atômico do nuclídeo-pai é reduzido uma unidade (Eq. 2.3) [2]. Tal como no decaimento β^- , o neutrino é necessário para conservar a energia da reação nuclear [5].



Captura Eletrónica (EC)

O decaimento por captura eletrónica compete com o decaimento β^+ para radionuclídeos ricos em prótons. Neste processo, um eletrão normalmente da orbital K, devido à sua proximidade com o núcleo, é capturado por um próton no núcleo, formando um neutrão e emitindo um neutrino, ν , para que haja conservação de energia. As radiações emitidas até o núcleo-filho passar do seu estado excitado para o seu estado fundamental, são fótons nucleares (raios-X), que correspondem à energia perdida pela passagem de um eletrão de uma orbital mais externa para uma mais interna onde se encontra o vazio deixado pelo primeiro eletrão (Figura 2.4) [2]. O processo geral da captura eletrónica é dado pela Eq. 2.4:

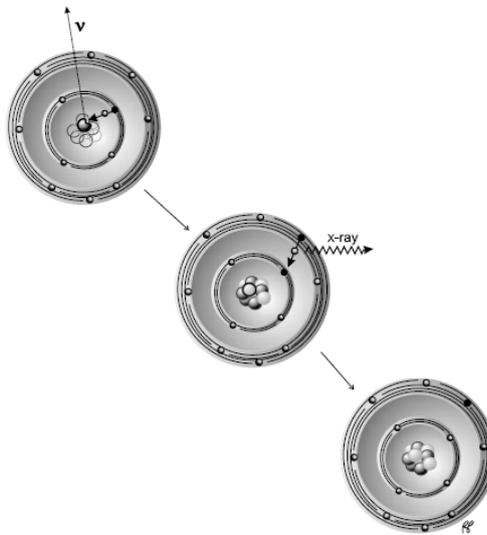
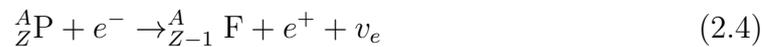


Fig. 2.4: Captura eletrónica [4].

2.1.4 Excesso de Energia

Emissão Gama

Halliday [1] afirma que a radiação gama foi descoberta em 1900 por Villard, sendo definida como uma radiação de alta energia, que é emitida sob a forma de radiação eletromagnética, devido ao excesso de energia no núcleo atômico. O tipo de decaimento radioativo mais comum é a emissão de raios gama (transição isomérica), em que o núcleo passa de um estado de energia excitado para um estado de energia mais reduzido, através da emissão de radiação eletromagnética, por exemplo os raios gama (fotão de energia). Este processo pode ser observado na Figura 2.5. A energia dos raios gama emitidos, corresponde à diferença entre as energias do estado final e inicial [4].

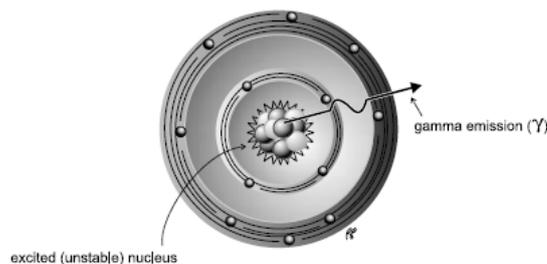


Fig. 2.5: Emissão Gama [4].

Conversão Interna

Neste processo, o núcleo excitado transfere a energia de excitação para um elétron numa orbital interna, normalmente a orbital K, o que provoca a sua ejeção do átomo, visto fornecer uma energia de excitação superior à energia de ligação do elétron. O elétron ejetado é designado por elétron de conversão e transporta uma energia cinética correspondente a $E_\gamma - E_b$, onde E_γ é a energia de excitação e E_b é a energia de ligação do elétron. A conversão interna deixa o átomo com um vazio correspondente ao elétron que foi ejetado de uma das suas orbitais. Este é preenchido por um elétron da orbital seguinte mais externa. Portanto, como é possível observar na Figura 2.6 a diferença energética entre estas duas orbitais é mostrada sob a forma de raio-X (A) ou um elétron livre (B) - elétron de *Auger* ou elétron secundário [7, 4]. A Eq. 2.5. é representativa deste tipo de processo.

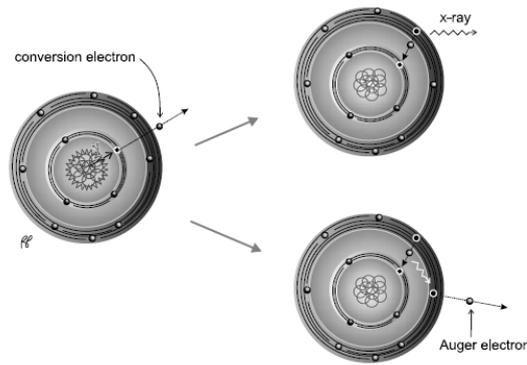


Fig. 2.6: Conversão interna [4].

As radiações ionizantes criadas por cada um destes processos podem ser divididas em dois tipos, radiações corpusculares (alfa, beta, nêutrons e prótons) e radiações eletromagnéticas (raios gama e raios-X) (Tabela 2.1).

Tab. 2.1: Propriedades das partículas incluídas nos vários tipos de decaimento radioativo, adaptado de [4].

Nome	Símbolo	Carga
Neutrão	n^0	Sem carga
Protão	p^+	Positiva (+)
Elétrão	e^-	Negativa (-)
Partícula beta menos (elétrão)	β^-	Negativa(-)
Partícula beta mais (positrão)	β^+	Positiva(+)
Raio gama (fotão)	γ	Sem carga
Raio-X	raio-X	Sem carga
Neutrino	ν	Sem carga
Antineutrino	$\bar{\nu}$	Sem carga

2.2 Interação da Radiação com a Matéria

Radiação eletromagnética de partículas pode interagir com átomos. O processo inicia-se com a transferência de energia de radiação para os átomos e moléculas, aquecendo a matéria ou mesmo modificando a sua estrutura. É ainda importante referir que as radiações eletromagnéticas (raios γ e raios-X) são também conhecidas como radiações penetrantes e as radiações de partículas (α e β) são conhecidas como radiações não-penetrantes. Tendo em conta este facto, os mecanismos de interação com a matéria podem ser separados em dois tipos: interação dos fotões com a matéria e interação de partículas carregadas com a matéria [2, 8].

2.2.1 Interação dos fotões com a matéria

Mecanismos de Interação de fotões com a Matéria

Devido à sua capacidade de passar através da matéria, os fotões interagem com os átomos. O tipo de interação pode ser caracterizada em função da energia dos fotões e do número atómico Z dos elementos que fazem parte da matéria. Existem três tipos de interações de fotões com a matéria: Espalhamento de *Compton*, Efeito fotoelétrico e Produção de pares [8]. O espalhamento de *Compton* constitui o tipo de interação dominante em materiais com um número atómico baixo, como por exemplo o tecido humano. O efeito fotoelétrico ocorre normalmente em materiais com um número atómico alto. A produção de pares acontece apenas com fotões de alta energia (superiores a 1020 keV) [4]. O gráfico apresentado na Figura 2.7 representa de forma simples a relação entre os vários tipos de interação, consoante o número atómico do átomo e a energia do fotão incidente [2].

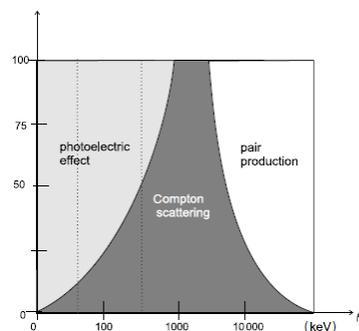


Fig. 2.7: Gráfico representativo da interação dos fotões com a matéria [4].

- Espalhamento de *Compton*

O fóton (raio γ) colide com um elétron da camada mais externa do átomo e transfere apenas uma parte da sua energia para esse elétron que é, por isso, ejetado para o exterior do átomo: elétron de *Compton* [3]. A energia do fóton é reduzida e a sua direção é alterada - espalhamento de Compton - e o ângulo formado por essa alteração depende da quantidade de energia transferida do fóton para o elétron (Figura 2.8) [4].

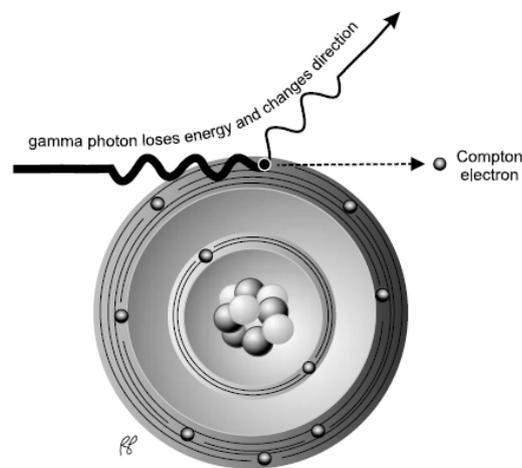


Fig. 2.8: Espalhamento de *Compton* [4].

- Efeito Fotoelétrico

Raios γ de baixa energia podem transferir a sua energia cinética para um elétron, normalmente da camada mais interna do átomo. Aquando da colisão do fóton com o elétron, este é ejetado do átomo - fotoelétron - (Figura 2.9) com a mesma energia do fóton incidente, já diminuída pela energia de ligação do elétron (Eq. 2.6, em que E_e corresponde à energia do elétron, E_f à energia do fóton e E_b à energia de ligação). O espaço vazio deixado pelo elétron ejetado é ocupado por um elétron de uma camada mais externa, sendo o excesso de energia emitido sob a forma de raio-X característico [3].

$$E_e = E_f - E_b \quad (2.6)$$

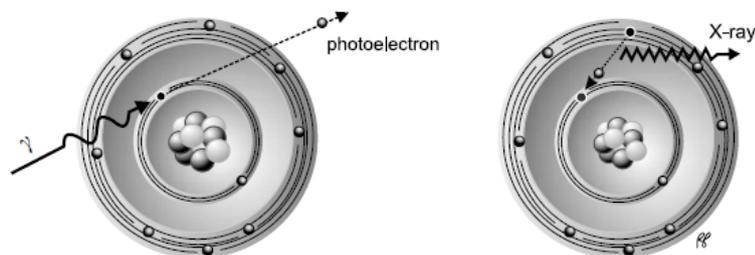


Fig. 2.9: Efeito Fotoelétrico [4].

- Produção de Pares

Quando a energia de um fóton é superior a 1.02 MeV, este pode interagir com o núcleo do átomo quando passa por este. Esta interação leva à produção de dois leptões com cargas opostas (ou seja, um elétron e um positrão) com a energia cinética restante do processo. Eventualmente, o positrão poderá sofrer uma aniquilação para produzir dois fótons de 511 keV, emitidos em direções opostas (Figura 2.10) [4].

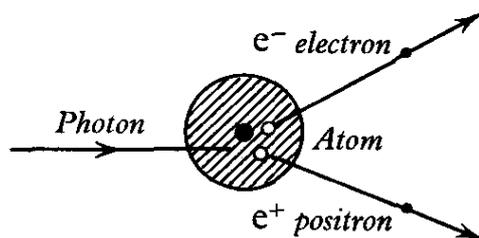


Fig. 2.10: Produção de pares [7].

Atenuação dos fótons na matéria

Como afirma Powsner [4], devido a constantes interações de fótons com a matéria, a intensidade de um feixe de radiação é reduzida. Por outras palavras, ao longo da passagem do feixe pela matéria, o número de fótons do feixe diminui. A esta perda de fótons dá-se o nome de atenuação. Fótons de baixa energia sofrem uma maior atenuação, contrariamente aos fótons de maior energia que sofrem uma menor atenuação, como é possível observar na Figura 2.11. O processo de atenuação inclui os três tipos de interação com a matéria já referidos: Efeito de *Compton*, efeito fotoelétrico e produção de pares. Na Eq. 2.7, é apresentada a expressão que descreve a atenuação de um feixe de fótons com uma determinada intensidade inicial I_0 ,

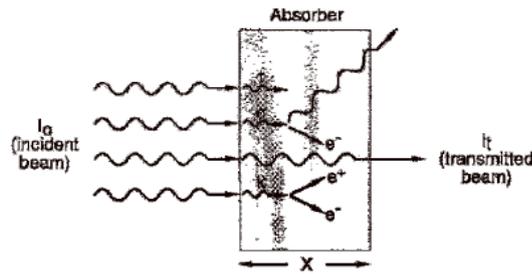


Fig. 2.11: Atenuação dos fótons na matéria [2].

que passa através de um material absorvente de espessura x , sendo posteriormente transmitida com uma intensidade I_t . A constante μ representa o coeficiente de atenuação linear do material que absorve o fóton [4].

$$I_t = I_0 e^{-\mu x} \quad (2.7)$$

2.2.2 Interação de Partículas Carregadas com a matéria

Partículas carregadas, como por exemplo partículas alfa, beta e prótons, podem interagir com átomos enquanto passam através da matéria. Como consequência, poderá resultar em excitação ou ionização dos átomos. Relativamente à excitação, a radiação interage com o átomo e transfere total ou parcialmente a sua energia para os elétrons presentes no átomo, no entanto, a energia transferida não é suficiente para ejetar os elétrons das suas orbitais, pois é inferior à energia de ligação do elétron. No que diz respeito à ionização, as radiações que colidem com os átomos-alvo conseguem arrancar elétrons para o exterior do átomo, pois a sua energia supera a energia de ligação do elétron. Estes elétrons ejetados podem ser designados por elétrons primários e têm a capacidade de produzir mais excitação ou ionização no átomo [2].

Radioproteção

As radiações ionizantes, devido à elevada energia que possuem, têm a capacidade de penetrar na matéria, ionizando os átomos, rompendo ligações químicas e causando danos nos tecidos biológicos. É relevante voltar a frisar que a radiação ionizante tem uma utilidade prática, pois possui diversas aplicações na área médica, mais especificamente na área do diagnóstico médico e terapia; como por exemplo a radiologia, radioterapia e medicina nuclear, tornando-se uma solução bastante viável no que diz respeito a estas áreas. A título de exemplo, permite a destruição de células cancerígenas, curando ou minimizando a doença [9]. Apesar da vantagem associada às radiações ionizantes e do seu papel quase imprescindível na medicina moderna, a exposição do Homem a doses elevadas de radiação pode levar à destruição de células do corpo humano, alterações no ADN, bem como à indução de cancro. Este capítulo 3 deve assim a sua existência para demonstrar a forma como esta situação poderá ser contornada, através do uso de detetores de radiação ionizante. Deste modo, é inicialmente abordada a questão dos efeitos biológicos da radiação na secção 3.1, sendo também apresentados vários tipos de detetores gasosos na secção 3.2 e, por fim, na secção 3.3 é descrito o detetor de *Geiger-Müller*.

3.1 Efeitos Biológicos da Radiação

Para o objetivo deste trabalho é relevante desenvolver uma abordagem acerca da interação da radiação com os tecidos biológicos. Estas interações promovem efeitos biológicos adversos que provêm de uma exposição prolongada ou não à radiação ionizante. O problema da interação da radiação com as células está relacionado com a molécula de ADN, inserida no núcleo da célula. Esta interação modifica

a sua estrutura, provocando danos irreparáveis, o que promove a alteração da sua funcionalidade.

3.1.1 Classificação

Os efeitos biológicos da radiação podem ser distinguidos e classificados em Estocásticos ou Determinísticos.

Efeitos Estocásticos

São os efeitos biológicos que aumentam a sua probabilidade de ocorrência com o aumento da dose. É de realçar que a gravidade dos efeitos não aumenta com doses elevadas de radiação. Não existe um limiar (*threshold*), já que o dano de uma célula ou várias pode, de igual forma, conduzir ao aparecimento de uma anomalia. Estes são, portanto, os efeitos com mais risco para a saúde, devido ao facto de até baixas doses serem capazes de produzir danos graves. Doenças hereditárias e incidências de cancro são exemplos de efeitos estocásticos [6, 2].

Efeitos Determinísticos

Os efeitos determinísticos diferem dos estocásticos, no sentido em que a gravidade dos efeitos cresce com o aumento da dose de radiação. São então necessárias doses bastante elevadas para produzir estes efeitos, existindo um limiar de dose, no qual todas as doses inferiores a esse valor não produzem qualquer efeito. Cataratas, eritema e fibrose constituem exemplos deste tipo de efeitos [6, 2].

3.1.2 Modos de Ação

Como já foi referido, quando a radiação incide num tecido biológico, esta provoca danos nas células, mais propriamente na molécula de DNA devido à sua alta sensibilidade às radiações ionizantes. Os danos podem ser causados direta ou indiretamente.

Ação Direta

Como o próprio nome indica, a radiação consegue atingir a molécula de ADN diretamente, rompendo a sua estrutura molecular, como é exemplificado no esquema da Figura 3.1 (a). Esta alteração na estrutura da molécula de ADN promove danos celulares ou até mesmo a morte da célula. As células que sofrem esta exposição

podem, no entanto, sobreviver, existindo uma forte possibilidade sofrerem mutações, induzindo cancros ou outras anomalias. Este processo é mais comum quando se trata de radiações do tipo LET (*Linear Energy Transfer*), como partículas α ou neutrões e doses elevadas de radiação corpuscular [2].

Ação Indireta

Contrariamente à situação anterior, a radiação atinge as moléculas de água existentes na célula, que correspondem a 70% da sua constituição. Esta interação promove a produção de radicais livres potencialmente perigosos, tais como o peridroxil (HO_2) e o alcóxi (RO_2). Os radicais livres caracterizam-se por conterem um número de eletrões ímpar, tornando-os muito reativos; o que promove uma reação com as moléculas de ADN, causando danos na sua estrutura idênticos aos referidos para a ação direta (Figura 3.1 (b)). É ainda importante realçar que grande parte dos danos provocados pela radiação ionizante resultam do mecanismo de ação indireta, devido ao grande conteúdo em água das células, conferido anteriormente [2].

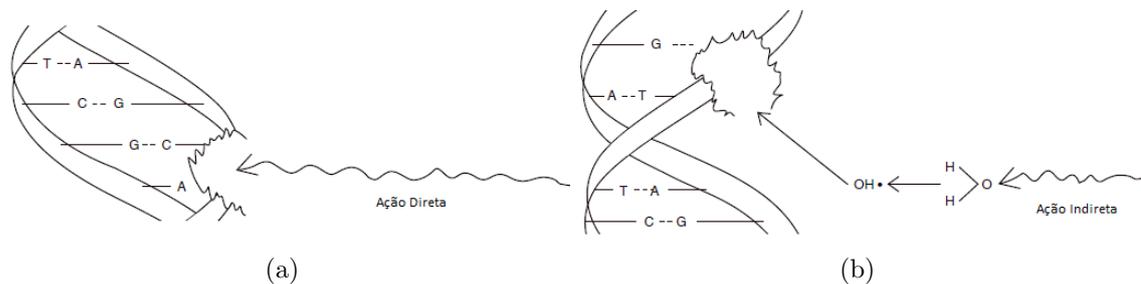


Fig. 3.1: Ação da radiação ionizante na estrutura da molécula de ADN: (a) Ação direta; (b) Ação indireta [2].

3.1.3 Tipos de Efeitos

Os efeitos biológicos podem ainda ser distinguidos consoante o tempo de aparecimento dos sintomas provocados pela interação da radiação com os tecidos biológicos.

Efeitos a Curto Prazo

Para compreender os efeitos a curto prazo é necessário introduzir o termo 'dose aguda', que corresponde ao nível de dose a que um determinado corpo é exposto, num curto espaço de tempo. Quando a quantidade de radiação for elevada o suficiente, doses agudas resultam em efeitos que se manifestam num período de horas ou

dias. Nesta situação, o período de latência, ou seja, o tempo decorrido entre a exposição à radiação e o aparecimento dos sintomas, é relativamente reduzido e torna-se progressivamente mais curto com o aumento do nível da dose de radiação [2, 10].

Efeitos a Longo Prazo

Os efeitos a longo prazo manifestam-se anos depois da exposição à radiação. O período de latência é, portanto, mais longo do que o anterior. Estes efeitos podem resultar de várias exposições a curto prazo, altas doses de radiação ou exposições crónicas de baixo nível ao longo de um determinado período de tempo. Assim, os efeitos a longo prazo são mais preocupantes que os de curto prazo. Estes podem ainda ser divididos em genéticos e somáticos. Relativamente aos efeitos genéticos, são as células reprodutoras que sofrem uma mutação no seu ADN, afetando apenas os descendentes e não o indivíduo exposto à radiação. Podem mesmo tornar-se letais para os descendentes ou provocar deficiências físicas ou mentais. Por outro lado, os efeitos somáticos afetam diretamente o indivíduo exposto à radiação e dependem de fatores como a área e volume do corpo atingido, profundidade, dose recebida e tempo de exposição. Estes levam ao aumento da probabilidade de cancro, cataratas ou redução do tempo de vida, entre outros efeitos [2, 10].

3.2 Detetores de fase gasosa

Considerando o âmbito da dissertação, serão apenas abordados os detetores intitulados '*Gas-filled detectors*', isto é, os detetores de fase gasosa, com maior relevância para o contador de *Geiger-Müller*, apresentado na secção 3.3. De um modo geral, os detetores de ionização funcionam com base na medição dos produtos secundários resultantes da interação da radiação com o material detetor [4].

Antes de especificar o funcionamento de cada detetor do tipo gasoso individualmente, é importante considerar o modo de funcionamento geral destes detetores. É ainda importante referir que as moléculas de qualquer gás se encontram relativamente distantes umas das outras e, portanto, estão mais suscetíveis de ser ionizadas por partículas muito carregadas, tais como α ou β , do que por radiação γ ou fótons. Este é o motivo pelo qual os detetores gasosos são utilizados, essencialmente, para deteção de partículas α e β , apesar de ser também possível a deteção de radiações γ e fótons. Estes detetores são normalmente constituídos por um cilindro com uma parede condutora, que corresponde ao elétrodo negativo (cátodo) e a sua parte central

ao eletrodo positivo (ânodo). No interior do cilindro existe um gás inerte (como por exemplo o argônio ou xénon), que por sua vez interage com a radiação, provocando a ionização das suas moléculas, seguido da geração de uma corrente elétrica com a aplicação de uma tensão entre os eletrodos. A passagem de um feixe de radiação ionizante pelo gás no interior no cilindro, conduz à ionização das moléculas gasosas e, como consequência, à produção de pares iônicos. Quando é feita a aplicação de uma tensão entre os eletrodos, os eletrões dirigem-se para o ânodo e os iões positivos (átomos ionizados), mais lentamente, para o cátodo, produzindo uma corrente que é posteriormente quantificada (Figura 3.2) [4].

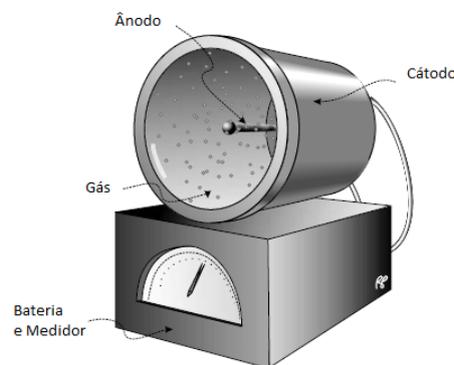


Fig. 3.2: Representação geral de um detetor gasoso [4].

3.2.1 Regiões de Operação dos Detetores Gasosos

A resposta dos detetores varia com a tensão aplicada entre os eletrodos. Desta forma, é possível criar quatro regiões principais em função da diferença de potencial aplicada, ilustradas na Figura 3.3 [11]. Quanto maior a tensão aplicada, menor será a probabilidade de recombinação dos pares iônicos, o que por sua vez gera uma corrente elétrica mais intensa.

- Região I (Região de recombinação): Para tensões muito baixas, os pares iônicos não recebem aceleração suficiente para alcançar os eletrodos, voltando a recombinar-se. Como resultado, as moléculas iniciais são reconstituídas, em vez de ser coletadas pelos eletrodos [12].
- Região II (Região de ionização): A partir de um determinado valor de tensão, a maior parte dos pares iônicos formados atingem os eletrodos e após coletados todos os pares, a corrente elétrica já não é aumentada. No caso de um evento

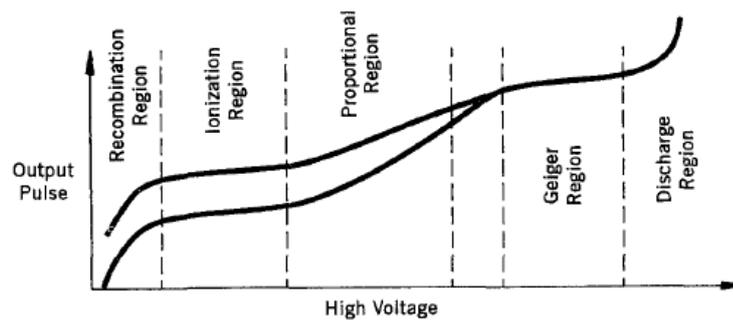


Fig. 3.3: Regiões de operação dos detectores gasosos [12].

individual, a corrente elétrica produzida é muito reduzida. Por este motivo, os detectores que se inserem nesta região de operação não são aconselháveis para eventos individuais. Um dos tipos de detectores que operam nesta região é a câmara de ionização [4].

- Região III (Região proporcional): Esta região caracteriza-se por um aumento substancial da tensão, o que conduz ao desenvolvimento de um fenómeno intitulado: 'Amplificação de gasosa'. Este aumento da tensão, promove também o aumento da aceleração dos eletrões que possuem uma energia cinética suficiente de modo a ionizar outras moléculas do gás (Figura 3.4). A este processo é dado o nome de avalanche ou avalanche de *Townsend*. Como tal, o valor na saída do processo é proporcional ao número de ionizações primárias. Após esta região surge uma região de proporcionalidade limitada, que é caracterizada por já não manter a proporcionalidade conseguida na região III [11, 4].

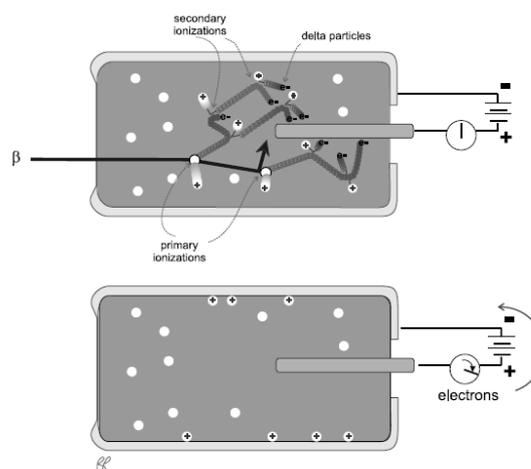


Fig. 3.4: Fenómeno de amplificação do gás de deteção [4].

- Região IV: Com um aumento superior de tensão, a partir de um determinado valor, a amplificação é maximizada, isto é, o processo de avalanche atinge o seu máximo. Deste modo, a intensidade da corrente é independente do número de iões produzidos pelos eventos iniciais. Quando os eletrões são fortemente energizados e atingem o ânodo com uma força relativamente elevada e até toda a carga ser recolhida, o detetor pára de detetar mais eventos. Os detetores característicos desta região são os detetores *Geiger Muller*. Superiores a esta região (Região de descarga contínua), os detetores não operam [4, 13].

3.2.2 Câmara de Ionização

A estrutura de uma câmara de ionização na sua forma mais simples é semelhante à estrutura básica dos detetores de fase gasosa, constituídos por gás de deteção que preenche a câmara, uma janela, um fio metálico no seu centro, um galvanómetro, responsável por medir a corrente gerada, e uma bateria, que mantém um potencial positivo no fio central (ânodo) e negativo nas paredes da câmara (cátodo) (Figura 3.2). Além disso, neste tipo de detetores a tensão é relativamente reduzida, sendo por isso incluídos na região II, isto é, região de ionização. Permitem a deteção de partículas α , β , raios γ e raios-X, sendo que, para as duas últimas, a sua eficiência é bastante baixa, normalmente inferior a 1%. A câmara de ionização não permite a medição de eventos individuais, mas sim a contagem da média de número de ionizações que ocorrem no seu interior [5, 2, 14].

Como defende Das e Ferbel [12], as câmaras de ionização não são muito sensíveis a pequenas variações de tensão o que, por sua vez, promove uma resposta de saída linear para uma gama ampla de sinais de entrada. Ainda assim, de um modo geral, a sensibilidade da câmara é também determinada pela sensibilidade do dispositivo de medição a esta associada. Calibradores de dose (*Dose Calibrator*), monitorizadores de radiação (*Survey Meter*) e dosímetros pessoais (*Pocket Dosimeter*) são as câmaras de ionização mais comuns, utilizadas na área da Medicina Nuclear [4, 2, 15].

Calibrador de dose

Este tipo de detetor é considerado uma câmara de ionização de mesa (Figura 3.5 (a)), isto é, funciona como um detetor utilizado para realizar medições rápidas para confirmação da quantidade correta de atividade numa dada amostra, como por exemplo, num radiofármaco a ser administrado. É constituído por uma câmara de forma

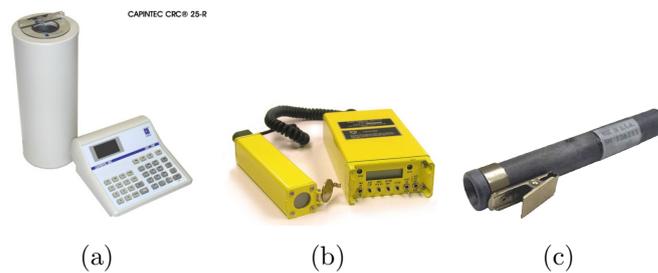


Fig. 3.5: Tipos de câmaras de ionização: (a) Calibrador de dose; (b) Monitorizador de radiação; (c) Dosímetro pessoal; (fontes: <http://www.phoenixphysics.com/>; <http://www.orau.org/>; <http://deqtech.com/>).

cilíndrica, com uma cavidade central preenchida com argônio e vestígios de halogênio a alta pressão (entre 5 a 12 atmosferas).

A câmara de ionização possui no centro um cilindro oco permeável a fótons. A corrente produzida no circuito é proporcional ao número de ionizações primárias na câmara. A tensão de operação varia entre 150 V a 200 V. A quantidade de corrente é registrada como radioatividade em Bq ou Ci [2].

Monitorizador de radiação

É também constituído por uma superfície cilíndrica externa e um fio central, correspondentes aos elétrodos e utiliza o ar como gás de ionização, baseado no mesmo método de funcionamento anterior. Contrariamente ao detetor referido anteriormente, é utilizado para medição da exposição de radiação em *Roentgens* (R) por hora (R/h) [2]. Um exemplo pode ser observado na Figura 3.5 (b).

Dosímetro pessoal

Consiste num tubo cilíndrico, que na parte central do seu interior, contém um pequeno filamento de quartzo isolado das paredes da câmara de ionização, que permite posteriormente a medição da exposição à radiação em *Roentgens* (Figura 3.5 (c)). Estes dispositivos funcionam segundo o princípio do eletrómetro. Inicialmente, o dosímetro é carregado por uma fonte de alimentação externa e, nesta situação, a leitura encontra-se na origem da escala. Quando existe uma exposição à radiação, o filamento perde parte da sua energia, sendo essa perda proporcional à quantidade de radiação a que foi exposto o dosímetro, que é lida em mR. Estes detetores possuem a vantagem de ser relativamente baratos e podem medir fótons até centenas de mR [4, 2].

3.2.3 Contador Proporcional

Os contadores proporcionais são semelhantes no seu funcionamento às câmaras de ionização. No entanto, normalmente o gás de ionização é árgon, ou metano e árgon misturado com ar. A principal diferença está relacionada com a tensão mais elevada aplicada para estes detetores, que resulta na amplificação do gás, ou "Avalanche de *Townsend*", já que este se insere na região de operação III. Com a aplicação de uma tensão mais elevada, um eletrão adquire energia suficiente para ionizar outro átomo. Devido à amplificação do gás, a radiação detetada provoca pulsos de corrente, o que permite a contagem de eventos individuais, contrariamente à câmara de ionização. Este tipo de detetor é adequado para discriminar o tipo de radiação detetada, dado que o tamanho do pulso gerado permite definir o tipo de radiação. Por exemplo, as partículas alfa ionizam mais rapidamente que as beta, permitindo a criação de um pulso maior na corrente, distinguindo assim o tipo de partículas detetadas [5, 8].

3.3 Detetor de Geiger Muller

O detetor *Geiger-Müller* ou GM (Figura 3.6 (a)), também conhecido como contador *Geiger-Müller* foi desenvolvido em 1913 por Hans Geiger e, mais tarde, em 1928, aperfeiçoado com ajuda de Walther Müller. É um dos detetores do tipo gasoso mais utilizados, devido ao seu prático funcionamento, boa qualidade, facilidade de manutenção, elevada sensibilidade e baixo custo. De notar que o contador GM, como o próprio nome indica, efetua a contagem de eventos radioativos, não medindo a energia das partículas, nem distinguindo o tipo de radiação [5, 16, 17].

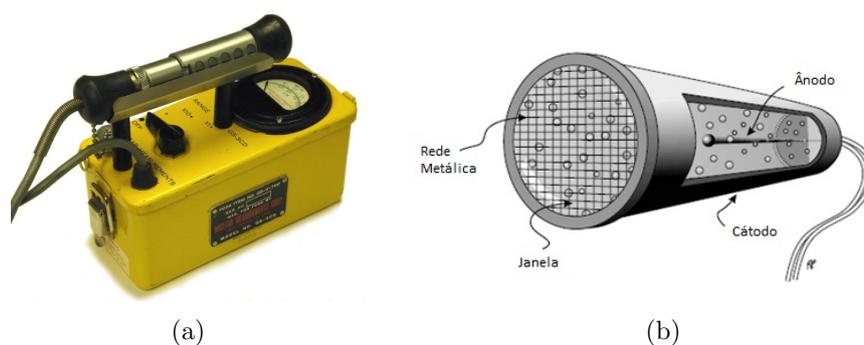


Fig. 3.6: Tubo de *Geiger-Muller*: (a) Exemplar comercial (fonte: <http://www.nuclearconnect.org>); (b) Representação esquemática (adaptado de [4]).

3.3.1 Estrutura

Na Figura 3.6 (b) é possível observar a estrutura geral de um tubo GM. Possui uma janela que permite a entrada da radiação proveniente de uma fonte radioativa, normalmente constituída por uma fina camada de mica e protegida por uma rede metálica. Em alguns tubos a janela é protegida por um invólucro de alumínio. No seu interior está presente um fio condutor metálico que funciona como ânodo e a parte interior do cilindro que constitui a câmara como cátodo. O gás utilizado normalmente para o tubo GM é o árgon, juntamente com vestígios de halogénio ou metano [4].

3.3.2 Princípios de Funcionamento

O detetor GM Consiste numa pequena câmara de ionização que opera com uma tensão relativamente alta (entre 400 V e 2000 V), região IV. Tal como nos princípios de funcionamento geral dos detetores gasosos, quando a radiação ionizante passa pela janela e entra no tubo GM, as moléculas do gás no seu interior são ionizadas, formando cargas positivas (iões) e negativas (eletrões), que se dirigem para o cátodo e para o ânodo, respetivamente, devido ao campo elétrico criado pelos elétrodos. Devido ao facto dos eletrões possuírem uma massa inferior aos iões, têm uma maior mobilidade, podendo adquirir uma energia cinética suficiente ao ponto de desenvolver ionizações secundárias, enquanto se dirigem para o ânodo. Dessa forma, provocam a amplificação das cargas elétricas no gás ou avalanche de *Townsend* – Região de proporcionalidade. Como a tensão aplicada ao tubo é bastante elevada, o número de cargas elétricas promove uma alteração considerável do campo elétrico no interior do tubo, deixando de existir uma proporcionalidade característica da região III. Surgem, então, uma série de avalanches consecutivas, que são desenvolvidas devido à emissão de fotões pelas moléculas do gás no interior do tubo, que são excitadas devido ao impacto dos eletrões. Por sua vez, esses fotões interagem com as paredes e gás do tubo, formando uma terceira geração de eletrões que espalham a descarga nas proximidades do elétrodo positivo, o ânodo. Este processo de sucessivas ionizações, que promove avalanches consecutivas, que ocorrem no interior da câmara, encontra-se esquematizado na Figura 3.7. Nesta fase, o processo já se encontra na região de operação IV ou de *Geiger Muller* [4, 5].

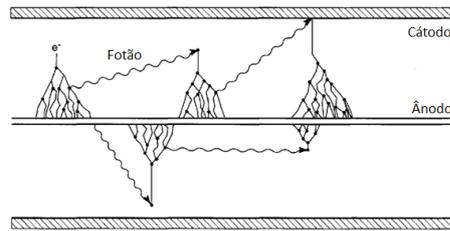


Fig. 3.7: Representação esquemática do fenômeno de avalanches de *Townsend* (adaptado de [5]).

Gás de Atenuação (*Quenching Gas*)

Para travar este processo de avalanches consecutivas, é utilizada a 'Técnica de *Quenching*' (Figura 3.8). Enquanto a descarga do tubo não for neutralizada por esta técnica, o gás de detecção fica saturado, não conseguindo contabilizar qualquer radiação que entre, posteriormente, pela janela. Esta inatividade de resposta (incapacidade) é conhecida como '*dead time*', isto é, tempo morto, que dura aproximadamente 100 a 500 μs . Esta técnica, típica da região de *Geiger-Müller*, consiste na adição de um gás de atenuação no interior na câmara de ionização, como por exemplo vapores orgânicos (etileno, álcool, xileno ou isobutano) ou gases de halogênios (cloro ou bromo). O objetivo deste gás é precisamente neutralizar os íons, por recombinação, quando estes se dirigem para o cátodo, o que reduz o fluxo de pares iônicos, inibindo assim o desenvolvimento das avalanches consecutivas [4, 5, 2].

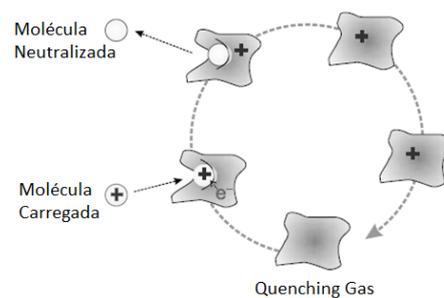


Fig. 3.8: Função do gás de quenching no interior do tubo (adaptado de [4]).

Implementação do sistema

Abordada a temática da radioproteção, pretende-se a implementação de um sistema que permita a proteção de um determinado ambiente, ou seja, a deteção de radiação ionizante, o que por sua vez pode prevenir eventuais exposições por parte da população. Para detetar a radiação é conveniente um sensor de funcionamento simples e com a capacidade para detetar pequenas e altas doses de radiação ionizante. O sensor utilizado no sistema foi um tubo GM, pelas várias características que apresenta, já referidas na secção 3.3. O sistema deve permitir a deteção de radiação ionizante num determinado espaço, gerar um sinal acústico quando a radiação é detetada, a leitura de contagens provenientes do sensor e, por fim, a transmissão dessa informação para um PC, via *wireless*. Este capítulo foi dividido em cinco secções, correspondentes aos vários aspetos relativos à implementação do sistema. Na secção 4.1, é apresentado o desenho do sistema, na secção 4.2 o circuito eletrónico implementado, na secção 4.3 os aspetos relativos à aquisição de dados e na secção 4.4 as formas de comunicação que se podem utilizar.

4.1 Desenho do Sistema

Para uma melhor organização na implementação do sistema foi essencial o desenvolvimento de um diagrama de blocos, que inclui os vários módulos que compõem o sistema. Este encontra-se representado na Figura 4.1 e é constituído por três módulos principais: circuito eletrónico, aquisição de dados e comunicação. O primeiro módulo foi dividido em três submódulos, correspondentes às diferentes partes que constituem o circuito. O módulo de aquisição de dados inclui todo o processo ligado à aquisição de dados, tendo sido realizado através da utilização de um conver-

ador analógico-digital (ADC). Por último, o módulo correspondente à comunicação diz respeito aos protocolos de comunicação usados no trabalho.

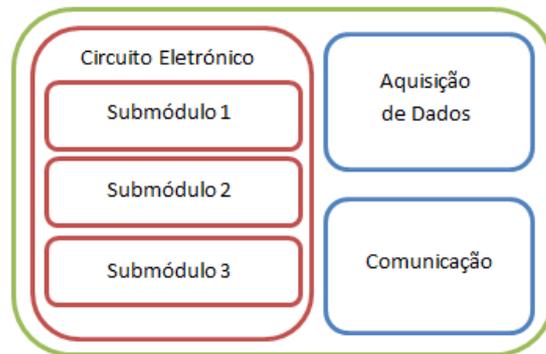


Fig. 4.1: Diagrama de Blocos.

4.2 Circuito Eletrónico

Esta secção está relacionada com uma parte do *hardware* que compõem o sistema desenvolvido, nomeadamente o circuito eletrónico de deteção. De um modo geral, o circuito permite a ligação de um tubo GM, que deverá ser devidamente alimentado, a transformação dos eventos captados pelo tubo em impulsos, que serão posteriormente contabilizados e um sistema acústico que ativa aquando da deteção de eventos no tubo GM.

Foram estudados vários circuitos eletrónicos para as finalidades apresentadas, optando-se pelo circuito apresentado na Figura 4.2. Neste, foram realizadas pequenas alterações, nomeadamente ao nível dos componentes utilizados, pelo facto de não ser possível a sua compra no site original¹. Antes de dar início à montagem do circuito numa *breadboard* foi feita uma listagem de todos os componentes necessários à sua realização, que pode ser consultada na Tabela 4.1. Na Figura 4.3 pode observar-se o circuito implementado na sua globalidade.

Foi ainda concebido um protótipo do circuito, através da solda dos componentes eletrónicos numa placa de prototipagem, para tal, foi necessário duplicar os componentes eletrónicos utilizados. O processo desenvolvido é apresentado no último tópico desta secção.

¹ <https://sites.google.com/site/diygeigercounter/>

Tab. 4.1: Componentes Eletrônicos para implementação do circuito de detecção

Componente	Quantidade
Circuito Integrado 555	1
Schmitt trigger	1
Resistência R1 220 k Ω	1
Resistência R2 3,9 k Ω	1
Resistência R3 330 Ω	1
Resistência R4=R9=R13 100 k Ω	3
Resistência R5 (Potenciômetro) 100 Ω	1
Resistência R6 1 M Ω	1
Resistência R7 4,7 M Ω	1
Resistência R8=R12 1,5 k Ω	2
Resistência R10 27 k Ω	1
Resistência R11 470 k Ω	1
Resistência R14 18 k Ω	1
Díodo D1=D3=D4=D5 Díodo 1N4148 (600 V)	4
Díodo D2 1N4937 (600 V) Fast Recovery	1
Díodo D0 MUR120	1
Condensador C1 100 μ F	1
Condensador C2 0,001 μ F	1
Condensador C3=C4 0,01 μ F/600v	2
Condensador C5 330 pF	1
Condensador C6=C7 0,1 μ F	2
Condensador C8 0,22 μ F	1
Transistor Q2 STX13005A	1
Transistor Q1=Q3 2N4401-704	2
Bobine 5–10 mH	1

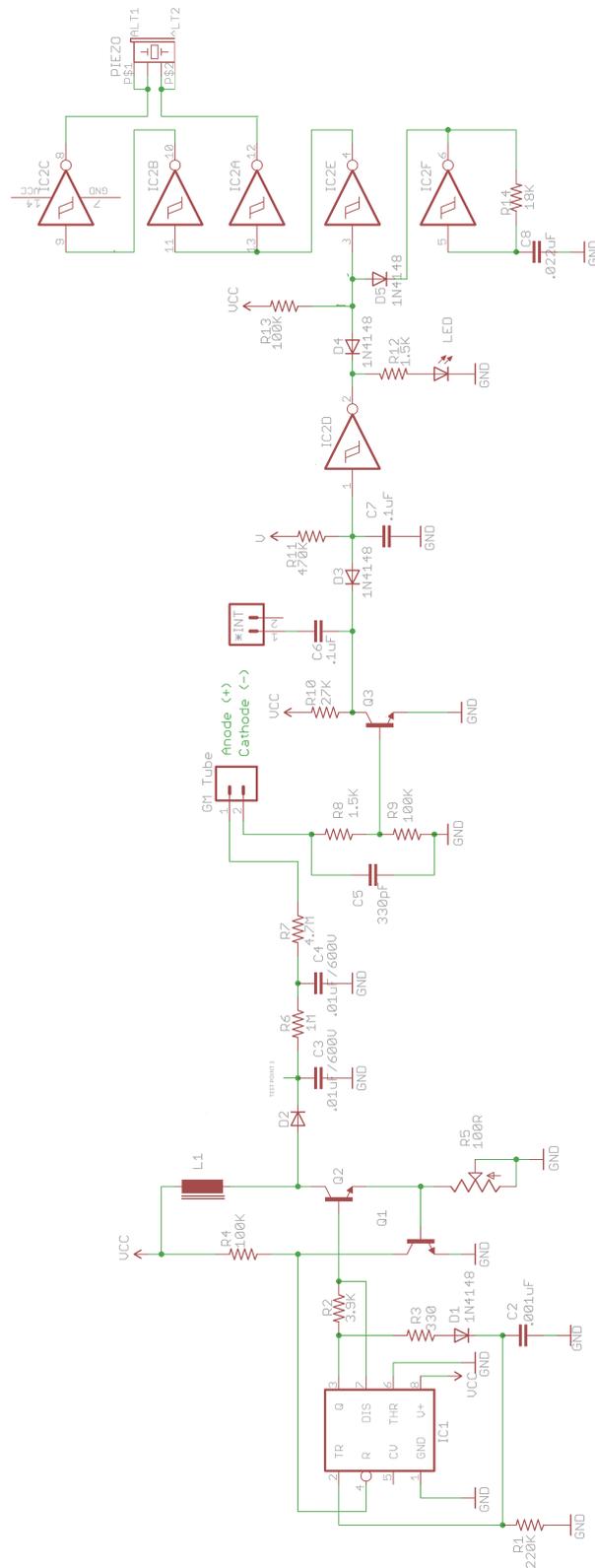


Fig. 4.2: Esquema do circuito eletrônico implementado.

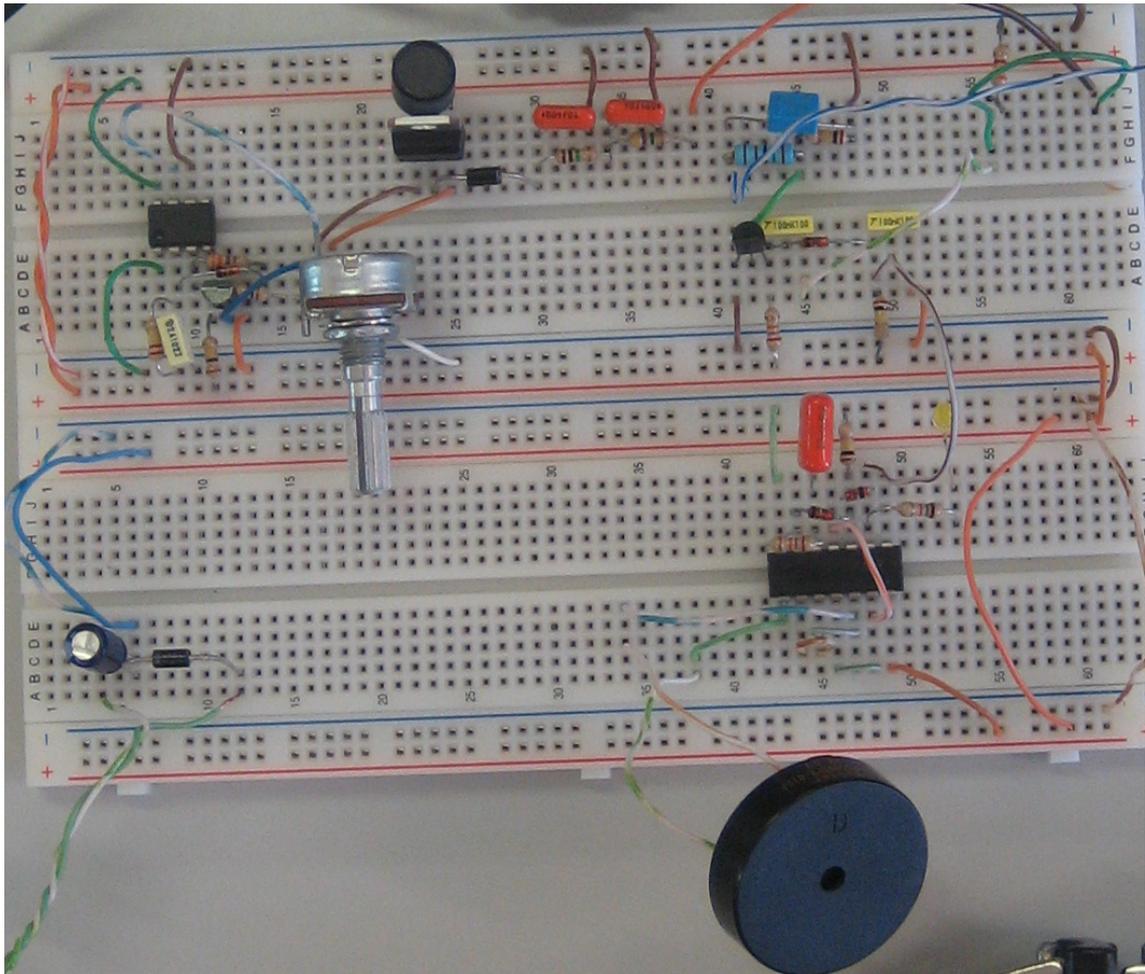


Fig. 4.3: Circuito eletrônico implementado.

4.2.1 Fonte de Alimentação

No que diz respeito à alimentação do circuito de detecção (Figura 4.4), são necessários 9 volts (V) de corrente contínua, que podem ser conseguidos através da utilização de uma pilha conectada ao circuito ou um transformador regulável ajustado para 9 V. O funcionamento deste bloco é assegurado por um condensador de 100 μF . Como afirma Horowitz [18], o condensador é um dos componentes geralmente essenciais para a construção de qualquer circuito electrónico. São, normalmente, constituídos por eletródos, existindo entre eles um material dielétrico (isolante).

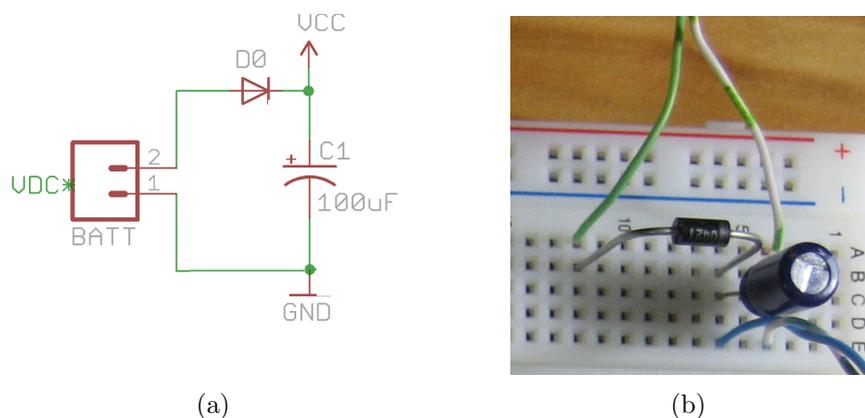
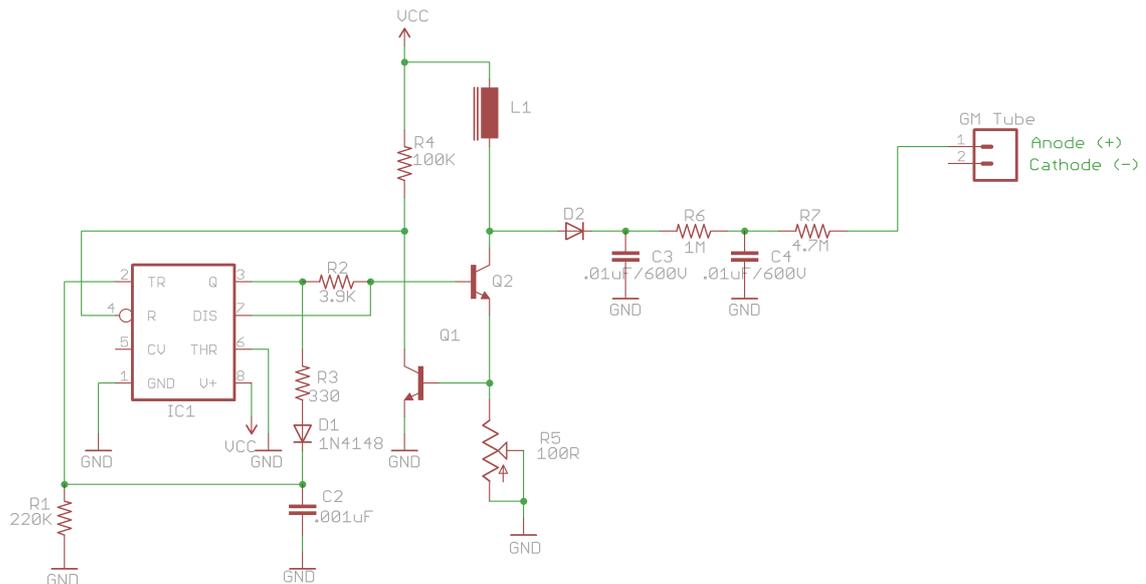


Fig. 4.4: Alimentação do circuito de detecção: (a) Esquema elétrico; (b) Implementação física.

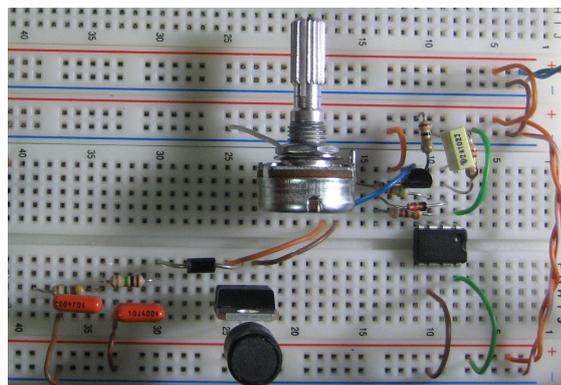
Para proteção do circuito de detecção contra, por exemplo, a troca de polaridade dos terminais da pilha ou transformador, é utilizado um diodo. O diodo é constituído por dois terminais, ânodo (+) e cátodo (-), e permite apenas que a corrente circule num único sentido, isto é, a corrente só consegue passar através do diodo quando é aplicada uma tensão positiva ao ânodo e uma tensão negativa ao cátodo [19]. Além disso, os diodos são por vezes utilizados como retificadores de corrente, permitindo mudar de corrente alternada (AC) para corrente contínua (DC) [18, 20]. Desta forma, é utilizado o diodo MUR120, que permite a averiguação deste tipo de acontecimentos, impedindo a passagem de corrente caso haja inversão da polaridade, protegendo desse modo o circuito. Este módulo fornece ao circuito de detecção a tensão indispensável para o seu adequado funcionamento.

4.2.2 Alimentação do Tubo GM

Esta subsecção refere-se à forma como é conseguida a tensão para alimentar o tubo GM. Para tal foi utilizado o circuito integrado (CI) 555. O esquema elétrico do circuito responsável por alimentar o tubo GM, pode ser observado na Figura 4.5 (a), bem como a sua implementação na Figura 4.5 (b).



(a)



(b)

Fig. 4.5: Circuito de alimentação do tubo GM: (a) Esquema elétrico; (b) Implementação física.

O CI 555 é um dos circuitos mais utilizados na atualidade devido às suas várias aplicações e configurações. A parte analógica do circuito é responsável pelo controlo do tempo em que o *output* do circuito se encontra ligado (ON) ou desligado

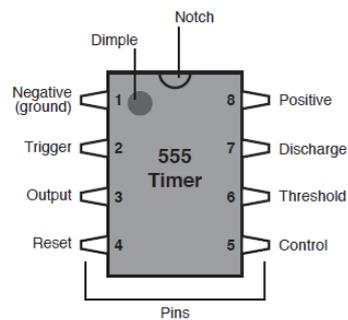


Fig. 4.6: Circuito integrado 555: Identificação dos pinos [21].

(OFF). Para esse efeito é necessário uma "rede" constituída por um condensador e uma resistência formando um circuito RC que, consoante os seus valores, determina o intervalo de tempo pretendido. Na Figura 4.6 é possível observar a correspondência de cada um dos pinos do CI 555, bem como a definição de cada pode ser encontrada na tabela 4.2.

Quando o *output* do CI está no nível "alto" (*High*), o condensador C2 (0,001 μF) é rapidamente carregado, a corrente passa pela resistência R3 de 330 Ω e pelo diodo D1 1N4148. O sinal de saída (pino 3) proveniente do CI 555 é apresentado na Figura 4.7 e corresponde a uma onda quadrática, o que comprova a funcionalidade do CI 555.

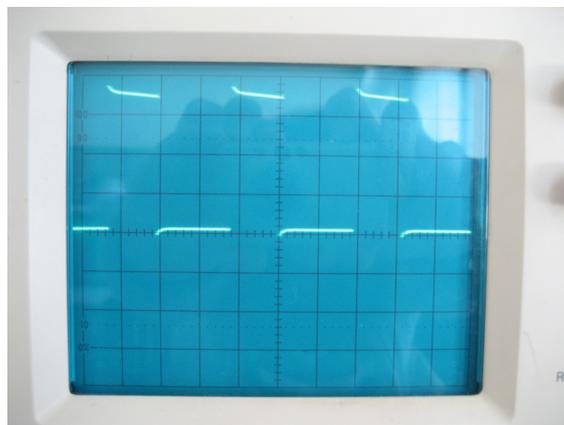


Fig. 4.7: Sinal de saída do pino 3 do CI 555.

O transistor, como definição geral, é um dispositivo eletrónico capaz de amplificar um sinal e produzir uma saída com uma tensão superior à da entrada [18, 20]. Na montagem deste trabalho, o transistor Q2 STX13005 é ativado, permitindo a passagem da corrente proveniente da bobine L1 (10 mH). A frequência do sinal proveniente do CI 555 faz a bobine oscilar aumentando a tensão e a passagem pelo potenciômetro R5 (100 Ω) permite ligar o transistor Q1 (2N4401). À medida que o

Tab. 4.2: Descrição dos pinos do CI 555 [19].

Pino	Descrição
Ground (1)	Ligação à massa (<i>ground</i> -GND)
Trigger (2)	Como o próprio nome indica, consiste num gatilho que permite iniciar o temporizador. O <i>Trigger</i> é do tipo baixo, isto é, o temporizador só inicia quando a tensão neste pino corresponder a $1/3$ de V_{cc} . Assim que o CI for ativado através deste pino, o <i>output</i> fica no modo alto;
Output (3)	O <i>output</i> está no nível baixo quando a tensão é aproximadamente 0 e alto quando o valor de tensão está próximo de V_{cc} . O valor exato da saída, ou seja, do tempo que está no nível alto ou baixo, depende das ligações dos restantes pinos constituintes do CI;
Reset (4)	Pode ser utilizado para reiniciar o 555. Tal como o <i>Trigger</i> , este pino é uma entrada do tipo baixo. Deverá estar ligado à tensão V_{cc} para o 555 funcionar;
Control (5)	Normalmente, este pino encontra-se ligado ao GND;
Threshold (6)	Este pino é responsável por controlar a tensão através do condensador que é descarregado através do pino 7. Quando esta tensão alcança $2/3$ de V_{cc} , o ciclo de tempo termina e o <i>output</i> no pino 3 passa ao estado baixo;
Discharge (7)	É utilizado para descarregar um condensador externo que trabalha juntamente com uma resistência para controlar o intervalo de tempo pretendido. Normalmente, este pino é conectado à tensão V_{cc} através de uma resistência e ao GND através de um condensador;
Vcc (8)	Ligação à fonte de alimentação (V_{cc}).

valor da resistência do potenciômetro varia, existe também uma variação de corrente na base do transístor Q1, ativando o seu funcionamento e amplificando o sinal.

Após vários testes, concluiu-se que o valor ideal para a resistência R5 seria de 6,5 Ω , de modo a permitir uma tensão suficiente para o funcionamento do tubo GM. É ainda importante referir que a resistência R1 de 220k Ω e o condensador C2 de 0,001 μ F estão conectados ao pino *threshold* do CI, permitindo o ajuste da frequência do CI². É também utilizado um diodo de recuperação rápida (*Fast Recovery Diode*), que contrariamente aos diodos comuns, tem uma resposta mais rápida e eficaz quando se funciona com valores de tensão muito elevados [22]. Devido à sua elevada capacidade de recuperação, evita que a corrente seja conduzida no sentido inverso, mesmo que por milissegundos [23].

Desta forma, no ponto de teste 1 observado na Figura 4.2 foi conseguida uma tensão de aproximadamente 540 V. Esta tensão é bastante elevada, sendo no entanto, essencial dado que após o diodo, surgem dois condensadores de 0,01 μ F/600 V que permitem o armazenamento de carga suficiente, por forma a alimentar adequadamente o tubo GM. Esta parte do circuito é então conectada ao ânodo (+) do tubo GM. Na Figura 4.8 podem ser observados os terminais de alimentação do tubo GM.



Fig. 4.8: Ligações ao tubo GM.

4.2.3 Produção do sinal analógico para aquisição e geração de um sinal acústico

Esta fase diz respeito ao submódulo correspondente à geração de um sinal analógico, em resultado da ionização do gás presente no tubo GM por partículas ionizantes. A avalanche de *Townsend* criada pela radiação primária, provoca uma breve descarga a partir do ânodo (+) para o cátodo (-)³. O circuito correspondente a este

² <https://sites.google.com/site/diygeigercounter/>

³ <http://www.m0dgq.co.uk/geiger%20counter.html>

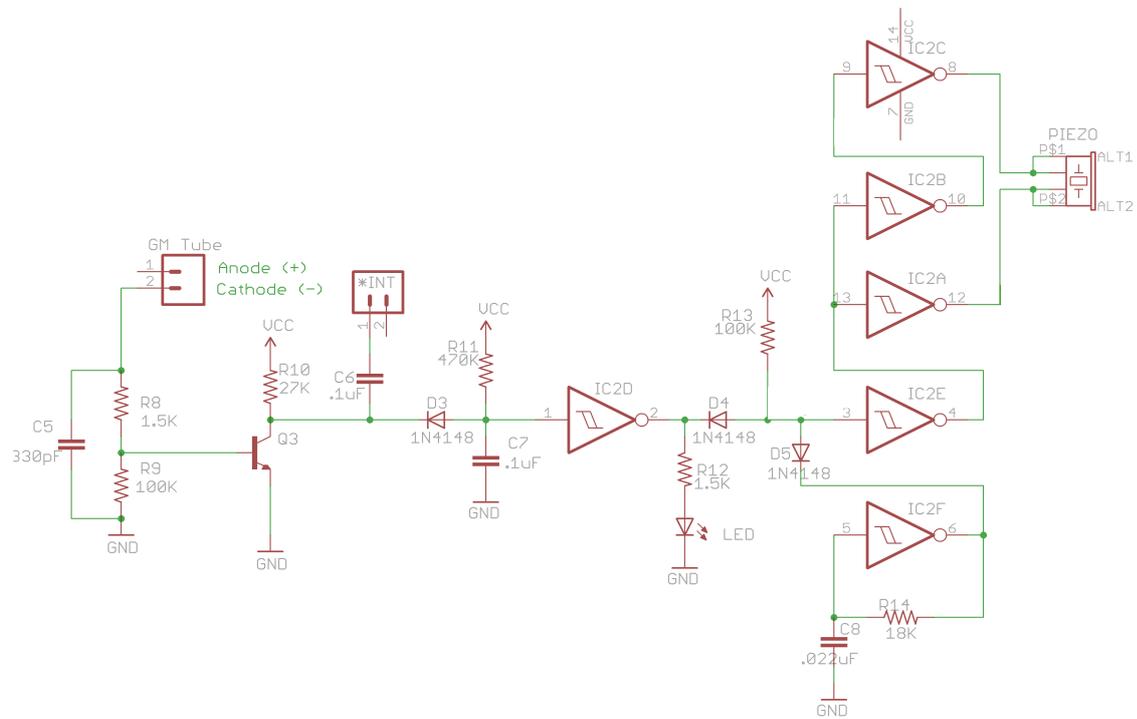
módulo encontra-se apresentado na Figura 4.9 (a), bem como a sua implementação física na Figura 4.9 (b).

O pulso de corrente proveniente do tubo passa por duas resistências, (R8 de 1,5 k Ω e R9 de 100 k Ω) ligadas em paralelo com um condensador de 330 pF diminuindo-se assim a sua intensidade, conectando o transistor Q3 (2N4401). Como tal, a corrente segue até ao condensador C6 (0,01 μ F), ligando por sua vez ao microcontrolador (Arduino), que posteriormente desempenhará a função de Conversor Analógico/Digital. Para além da funcionalidade mencionada, o circuito permite gerar um aviso acústico aquando da deteção de radiação ionizante. A resistência R11 de 100 k Ω e o condensador C7 de 0,1 μ F são responsáveis pelo ajustamento da duração de cada *click*, ou seja, de cada contagem detetada pelo tubo. Neste fase, é utilizado um circuito comparador do tipo disparador de *Schmitt* (*Schmitt Trigger*) para a geração de um sinal acústico, aquando da deteção de radiação ionizante pelo tubo GM. Um *Schmitt Trigger* funciona com base numa realimentação *feedback* positiva, que é introduzido por adição de parte da tensão de saída à tensão de entrada. É muitas vezes utilizado juntamente com outros circuitos, para gerar sinais padrão, como ondas quadráticas, triangulares ou pulsadas [24]. Outra das aplicações mais comuns dos *Schmitt Trigger* é a capacidade de "imunidade" ao ruído, permitindo limpar um sinal [25]. O impulso de saída proveniente de Q3 segue para o primeiro inversor do *Schmitt Trigger* 74ACT14N (pino 1) utilizado neste circuito, que pode ser observado na Figura 4.10. Outro inversor é usado para criar um oscilador de 260 Hz (um oscilador de 3,5 kHz foi utilizado para a frequência do "*piezo buzzer*"), através do uso do condensador C8 (0,022 μ F) e resistência R14 (18k Ω). Com recurso aos componentes R13 (100 k Ω) e D5 (1N4148), é obtida uma porta lógica NAND, que combina um pulso com o oscilador. No *piezo buzzer*, são ligados a um dos seus terminais dois inversores em paralelo para promover uma melhor condução do sinal. Desta forma, até à receção de radiação ionizante por parte do tubo GM, o sinal acústico é inexistente, tornando-se ativo assim que for detetado um evento, aumentando de intensidade consoante a proximidade da fonte radioativa⁴.

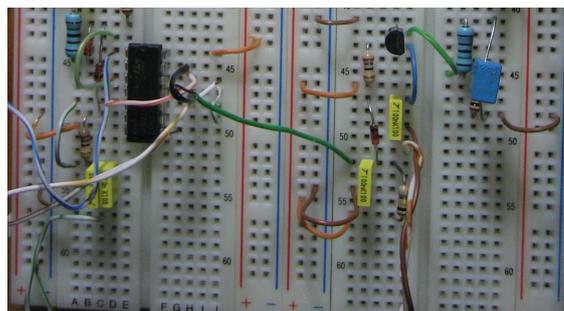
Ligação ao Arduino

Para a aquisição dos dados no PC é essencial a utilização de um conversor ADC, neste caso o Arduino, que será abordado em pormenor mais à frente. O circuito gera

⁴ <https://sites.google.com/site/diygeigercounter/>



(a)



(b)

Fig. 4.9: Circuito responsável pela produção do sinal analógico para aquisição e geração de um sinal acústico: (a) Esquema elétrico; (b) Implementação física.

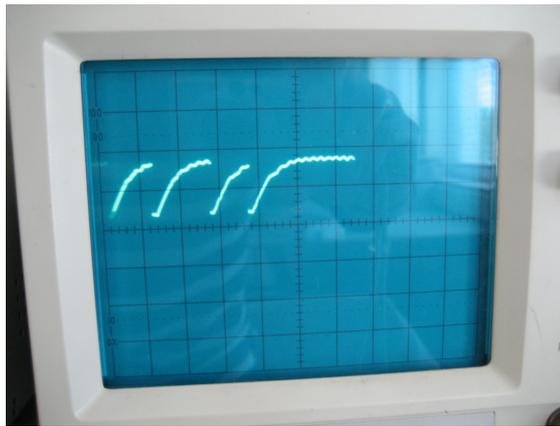


Fig. 4.10: Sinal de saída do pino 1 do *Schmitt Trigger*.

um sinal analógico, que será digitalizado pela placa Arduino e os dados e enviados para o PC.

Para iniciar a transmissão dos dados é necessário criar duas ligações: a primeira consiste na criação de uma ligação à massa GND comum entre a placa Arduino e o circuito implementado, ou seja, tornar a base do potencial elétrico comum entre os dois; a segunda diz respeito ao sinal analógico proveniente do circuito, em que a ligação é feita entre o circuito e uma das portas analógicas presentes na placa Arduino, neste caso a porta analógica 0.

4.2.4 Circuito Impresso PCB

No sentido de desenvolver um detetor de radiação ionizante mais robusto e estável, o circuito implementado numa *breadboard* foi soldado numa placa de prototipagem para que seja garantida a robustez das ligações. O circuito soldado obtido pode ser observado na Figura 4.11. Existe neste circuito um ponto de teste essencial para comparar o comportamento entre o protótipo e o circuito implementado na *breadboard*, que pode ser observado na Figura 4.2.

A soldadura dos componentes foi efetuada por partes, correspondentes aos módulos apresentados anteriormente. Após a soldadura do módulo responsável pela alimentação do circuito, foi possível verificar na placa, que as tensões de 0 (GND) e 9 V (V_{cc}) foram alcançadas. No módulo de alimentação do tubo GM, foi possível atingir valores elevados de tensão após o *Fast Recovery Diode* e variar a mesma com recurso ao potenciómetro. Após a soldadura do módulo correspondente à produção do sinal analógico para aquisição e geração de um sinal acústico, a montagem do

circuito ficou completa.

Apesar de o ponto de teste ter correspondido à tensão esperada, quando o tubo GM foi conectado ao circuito e testado com uma fonte de 3kBq , nenhum sinal acústico foi gerado, bem como não foi possível medir corretamente os eventos através do Arduino. Desta forma, foi possível concluir que poderá ter ocorrido alguma anomalia, aquando da soldadura do último módulo, já que os módulos anteriores funcionaram corretamente. Todas as ligações foram verificadas novamente, para garantir que as conexões entre os componentes foram realizadas de forma correta. Como tal, o problema poderá ter passado pelo próprio processo de soldadura. Este poderá estar relacionado com sobreaquecimento de componentes quando o ferro de soldar permanece por tempo excessivo junto dos terminais destes. Outro dos aspetos que poderá ter influenciado este resultado, foi o facto das pistas de ligação da placa, onde foram soldados os componentes, se danificarem com facilidade devido também ao excesso de calor produzido. Pelos motivos apresentados, é importante frisar que o processo de soldadura, representa um trabalho de grande precisão e rigor, já que qualquer erro poderá representar uma falha do *hardware*.

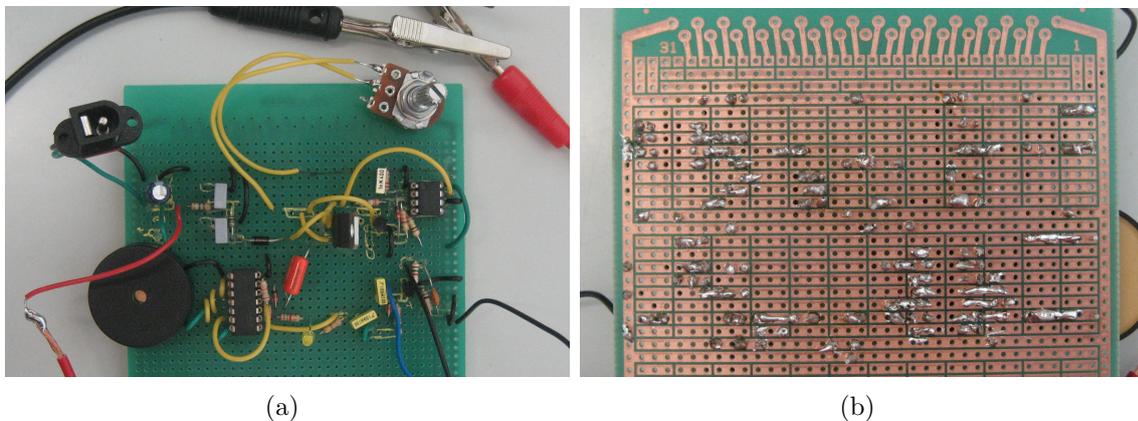


Fig. 4.11: Placa de prototipagem com os componentes soldados: (a) Vista de frente; (b) Vista posterior.

4.3 Aquisição de Dados

Após a implementação do circuito é necessário que os dados provenientes deste sejam enviados para um PC, por forma a registar a presença de radiação ionizante e quantificá-la. Os dados físicos que se pretende adquirir, provêm, como já foi referido

anteriormente, dos eventos ocorridos no tubo GM, em resultado da sua exposição a fontes de radiação. No âmbito deste trabalho foi usada uma fonte calibrada com 3 kBq de atividade. Este sinal é inicialmente do tipo analógico e deverá ser convertido num sinal digital, sendo para isso necessário um conversor analógico digital (ADC). Esta secção mostra a necessidade de utilização de um Conversor ADC para o processo de aquisição de dados e a escolha dos conversores ideais para os objetivos propostos.

4.3.1 Conversor Analógico-Digital

Nesta fase do trabalho foi utilizado um conversor ADC com o intuito de converter um sinal analógico proveniente do circuito eletrónico num sinal digital, de modo a recolher os dados, processá-los e armazená-los para consulta posterior. Um conversor ADC tem a funcionalidade de, a partir de um sinal analógico, neste caso a tensão proveniente do circuito, produzir um sinal digital. Através do esquema simplificado representado na Figura 4.12, é possível compreender o processo que ocorre quando um sinal é processado por um conversor deste tipo. Um sinal analógico, representado pela onda sinusoidal, passa pelo conversor e, em resultado, obtém-se um sinal digital representado por uma onda quadrática que varia entre 0 e 1. Como ferramenta para desempenhar a função de ADC, foi selecionada a plataforma Arduino numa primeira fase dos testes e, posteriormente, um módulo XBee para transmissão de dados via *wireless* [26].



Fig. 4.12: Representação esquemática de um conversor analógico-digital (ADC).

Conversores Analógicos-Digitais utilizados

O processo de conversão do sinal analógico para digital é indispensável quando se pretende adquirir dados provenientes do circuito eletrónico ou mesmo de dispositivos analógicos, como resistências variáveis, LDRs (Light Dependent Resistor) ou sensores de temperatura. Para a grande maioria dos testes de funcionalidade concretizados ao longo deste trabalho, foi necessária a utilização do conversor ADC em duas situações distintas. Nos testes iniciais não foram realizadas comunicações sem-fio (*wireless*), utilizando-se a plataforma Arduino para converter os sinais analógicos

em sinais digitais. Concluída esta fase, foram testadas as comunicações via *wireless*, em que o dispositivo escolhido para o papel de conversor ADC foi um módulo XBee.

Arduino

O Arduino é uma plataforma de prototipagem eletrónica *open-source*, baseada num *hardware* e *software* de baixo custo, de código livre e de fácil utilização. A plataforma Arduino é constituída por duas componentes: uma placa eletrónica (*hardware*) e um ambiente de desenvolvimento para criação de projetos (*software*). Pode ser utilizado para desenvolvimento de objetos interativos *standalone* ou para conectar o *software* no computador (como *Flash*, *Processing*, *VVVV*, etc.).⁵ ⁶ No que diz respeito ao *hardware* desta plataforma, de uma forma geral é constituído por entradas e saídas analógicas e digitais, por uma porta USB para comunicação em série e um microcontrolador responsável por executar as instruções dos programas desenvolvidos, instalados na memória física da placa Arduino. Este microcontrolador incorporado na placa é programado utilizando a linguagem de programação Arduino (baseado em *Wiring*) e o ambiente de desenvolvimento Arduino (baseado em *Processing*) [27, 28]. Desta forma, o programa instalado na placa Arduino dará ordens de aquisição e processamento dos dados e estes, já convertidos, serão lidos e mostrados no PC, através do ambiente de desenvolvimento integrado do Arduino [26, 29].

Existem no mercado vários modelos de placas Arduino, que podem ser utilizados (Duemilanove, Uno, Leonardo, Mega 2560, Mega ADK, Micro, Diecimila, Mini, Nano, Ethernet, Esplora, ArduinoBT, Fio, Pro (168), Pro (328), Pro Mini). Na Figura 4.13 (a) é apresentado o modelo UNO, uma das variantes utilizadas neste trabalho.

Na secção 5.1 serão apresentados os modelos empregues para funcionar como conversor ADC, bem como explicado o seu funcionamento, de um modo geral, identificados os pinos e uma descrição ao microcontrolador.

⁵ <http://arduino.cc/>

⁶ <http://www.kenleung.ca/portfolio/arduino-a-brief-history-3/>

Módulo XBee

Os módulos de comunicação XBee, também designados por rádios XBee, são dispositivos que funcionam através da transmissão de ondas rádio, permitindo a recepção e transmissão de dados, entre vários dispositivos deste tipo. É no microcontrolador integrado no módulo XBee que está inserido o protocolo *ZigBee*, descrito mais adiante. Este módulo de rádio-frequência foi desenvolvido pela *Digi International*⁷, que disponibiliza na sua plataforma *on-line* uma série de informações essenciais para a construção de uma rede deste tipo, bem como a apresentação e especificações dos vários produtos [30]. Existem pelo menos 30 combinações diferentes de *hardware*, *firmware*, protocolos, potências de transmissão e opções de antena, que dão origem aos vários módulos XBee disponíveis no mercado. Existem, no entanto, duas variações básicas dos XBee: Regular e a Pro, em que a diferença entre um e outro reside na potência de transmissão, sendo superior para o modelo Pro (68 mW). Os dados físicos provenientes de um dispositivo analógico quando passam pelo módulo XBee para transmissão, são automaticamente convertidos para o formato digital, estando o módulo devidamente configurado para essa função, como será explicado na secção 6.3. A secção 6.2 apresenta informações mais detalhadas sobre os módulos XBee, tais como os pinos que o constituem, a possibilidade de utilização de um *Shield* Arduino, antenas e adaptadores. Na Figura 4.13 (b), é apresentado um tipo de módulo XBee, para efeitos ilustrativos.

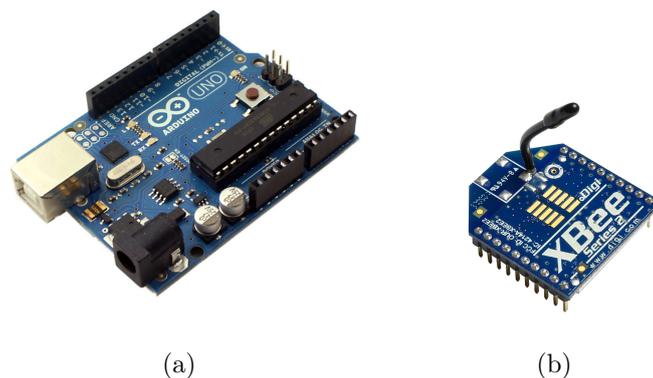


Fig. 4.13: Conversores analógicos-digitais utilizados: (a) Placa Arduino; (b) Módulo de comunicação XBee.

⁷ <http://www.digi.com/xbee/>

4.4 Comunicação de dados

Para que os dados físicos com origem no tubo GM sejam recolhidos num dispositivo externo torna-se necessária a comunicação entre os vários dispositivos. Para tal, nesta secção serão abordados vários aspetos relacionados com os tipos de comunicação disponíveis, o meio pelo qual a transmissão de dados pode ser feita e, por fim, o tipo de redes que se podem construir consoante o meio de transmissão de dados escolhido. No âmbito de cada um destes aspetos a ter em conta, será referenciada a escolha individualizada para cada um.

4.4.1 Tipos de Comunicação

Existem duas formas gerais para o envio de dados na área da ciência da computação: a comunicação em série e a comunicação em paralelo. Na comunicação em paralelo, grupos de bits são enviados simultaneamente, o que exige fisicamente a utilização de mais de que um fio para garantir a comunicação, o que promove a existência de ruído e utilização em distâncias curtas. Por oposição, na comunicação em série, é importante referir que o conceito '*Serial*' (*Serial Communication*) significa 'um após o outro', ou seja, a informação é enviada bit após bit. Esta forma de comunicação reduz significativamente os custos, devido ao facto de não serem necessários vários fios para a transmissão dos dados, permitindo ainda que esta seja feita através de distâncias mais longas⁸ [31]. A comunicação entre o Arduino e o XBee e entre o Arduino e o PC é em série. A Figura 4.14 representa esquematicamente a forma como a informação passa entre o Arduino e o computador, através do cabo USB. De salientar que o sinal ao passar no cabo USB já se encontra na forma digital, pois a conversão é realizada pelo conversor ADC, controlado pelo microcontrolador incorporado no Arduino.

4.4.2 Transmissão de Dados

A transmissão de dados em todos os meios de comunicação, passa por uma fase de codificação de uma amostra de dados sob a forma de energia. É sob essa forma de energia que os dados são transmitidos através de um meio de comunicação, como por exemplo infra-vermelhos, micro-ondas, satélite, fibra ótica, rádio, etc. Para a fase do trabalho relativa à comunicação entre dispositivos não conectados fisicamente

⁸ <http://www.pcs.usp.br/pcs2497/aula%20serial.pdf>

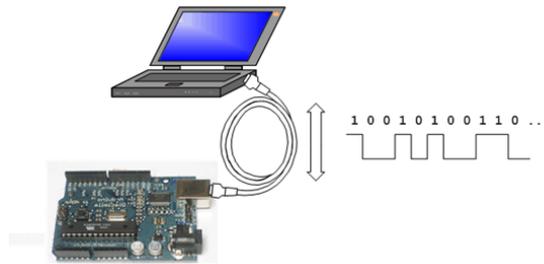


Fig. 4.14: Representação simplificada da comunicação em série entre o Arduino e o PC (fonte: <http://www.ladyada.net/learn/arduino/lesson4.html>).

apresentada no capítulo 6, o meio de transmissão utilizado foi por meio de ondas rádio. Este meio de transmissão consiste num sinal eletromagnético que se propaga num determinado espaço. Como tal, uma rede baseada neste tipo de transmissão funciona numa frequência específica para aquela rede, não necessitando de uma ligação física para transmissão de dados, podendo fazê-lo via *wireless*, como é o caso das várias redes desenvolvidas para os testes realizados. O dispositivo escolhido para a transmissão de dados via ondas rádio é o módulo XBee, já referido sucintamente na secção anterior [32, 33].

4.4.3 Redes Sem-fios

As redes sem-fios ou *wireless* são classificadas pelo IEEE (*Institute of Electrical and Electronics Engineers*) em quatro tipos principais: WPAN (*Wireless Personal Area Network*), WLAN (*Wireless Local Area Network*), WMAN (*Wireless Metropolitan Area Network*) e WWAN (*Wireless Wide Area Network*). Todas elas permitem comunicações via *wireless*, distinguindo-se, no entanto, no tamanho da rede. A rede utilizada para o presente trabalho enquadra-se nas WPAN, caracterizadas por pequenos alcances (10 a 100 metros). Aqui estão incluídas as tecnologias *Bluetooth*, especificadas pelo protocolo IEEE 802.15.1 e *ZigBee* especificada pelo protocolo IEEE 802.15.4 Os restantes tipos de redes são utilizadas para alcances de dimensões superiores, o que resulta num maior custo de implementação [34, 35, 36].

Protocolos de Comunicação

Existem três tecnologias *standard* distintas na atualidade, baseadas em comunicações via *wireless*: *Wi-Fi*, *Bluetooth* e *ZigBee*. Qualquer uma destas tecnologias

poderia ter sido escolhida para a implementação da rede de sensores de radiação ionizante. Contudo, foi essencial a realização de uma escolha viável de uma destas tecnologias, tendo em conta os recursos disponíveis. Na Tabela 4.3 encontram-se identificadas as principais diferenças entre estas tecnologias. O objetivo de um sistema deste tipo, em termos espaciais, passaria por uma zona constituída, por exemplo, por vários compartimentos, em que seria necessário colocar um sensor nas diferentes divisões, com um PC local na zona mais centralizada. Desta forma, o alcance exigido para a implementação de uma rede para esta finalidade não será muito elevado. Um dos factores decisivos para a escolha do tipo de rede é o baixo custo. O protocolo *ZigBee* oferece inúmeras vantagens no que diz respeito a custos, apresentando um consumo energético bastante reduzido, baixa complexidade e grande robustez. Além disso, foi concebido de raiz precisamente para aplicações de monitorização e redes de sensores. Comparativamente com o *Bluetooth*, o padrão *ZigBee* tem a possibilidade de trabalhar durante períodos de comunicação e um número de nós possíveis de implementar bastante superior [37, 38, 39, 40, 41].

Tab. 4.3: Comparação entre os protocolos de comunicação *Wi-Fi*, *Bluetooth* e *ZigBee* [37, 38, 39, 41].

	Wi-Fi	Bluetooth	ZigBee
Especificação da camada física	IEEE 802.11	IEEE 802.15.1	802.15.4
Alcance (m)	50 a 100	10 a 100	10 a 100
Frequência de Operação	2.4 e 5 GHz	2.4 e 5 GHz	868MHz na Europa e 2.4GHz no resto do mundo
Complexidade	Alta	Alta	Baixa
Consumo Energético	Alto	Médio	Baixo
Taxa de transferência (kbit/s)	11000	1000	250
Número Máximo de Nós	30	8	65000
Aplicações	Internet, transferência de ficheiros, vídeo e áudio	Substituição dos cabos entre dispositivos como telemóveis, PDAs, etc.	Controlo remoto, monitorização, sensores, etc.

F

Aquisição de Dados

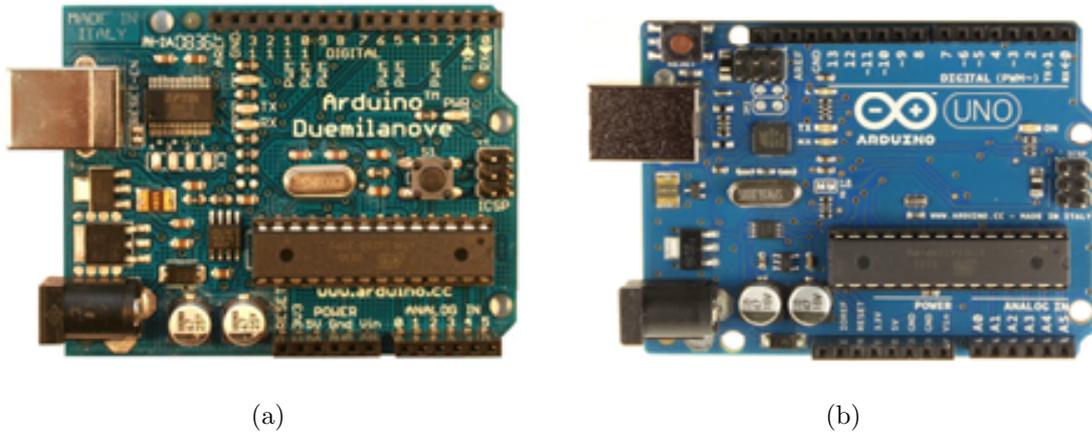
Como já foi referido anteriormente no capítulo 4, relativo ao desenho do sistema, um dos módulos está relacionado com a aquisição de dados que exige a utilização de um conversor ADC. Este capítulo pretende comprovar a utilização do Arduino como ADC, utilizado nos primeiros testes realizados para a aquisição de dados. Na secção 5.1 é abordada a questão da escolha dos Arduinos usados, comparação entre eles, funcionamento, características gerais, pinos e referência ao microcontrolador. Na secção 5.2 é dada uma descrição genérica dos programas utilizados e o seu respetivo funcionamento. Por fim, a secção 5.3 corresponde à descrição pormenorizada de alguns dos testes realizados para esta fase.

5.1 Especificações do Arduino

Para este trabalho foram utilizados dois modelos do Arduino: a placa Arduino Duemilnove, apresentado na Figura 5.1 (a), e a placa Arduino Uno, ilustrada na Figura 5.1 (b). As principais características destas placas encontram-se descritas na Tabela 5.1. De uma forma geral, estes dois modelos apresentam grande parte das suas características comuns, com a exceção do microcontrolador, memória *flash*, SRAM e EEPROM. A razão para o uso de dois modelos diferentes está relacionada com compatibilidade com o PC e com a facilidade de extrair o microcontrolador da placa; operação necessária numa fase posterior para programar os módulos XBee¹.

É essencial para a utilização do Arduino compreender as suas funcionalidades e constituintes da placa. Observando a Figura 5.2, é possível visualizar os vários constituintes da placa, devidamente identificados. A descrição exhaustiva dos

¹ <http://arduino.cc/>



(a)

(b)

Fig. 5.1: Modelos Arduino utilizados: (a) Arduino Duemilanove; (b) Arduino Uno (fonte: <http://arduino.cc/en/Main/Products>).

Tab. 5.1: Características dos modelos Arduino Duemilanove e Uno (PWM - *Pulse width modulator*).

Características	Arduino Duemilanove	Arduino Uno
Microcontrolador	ATmega168	ATmega328
Tensão de funcionamento	5 V	5 V
Tensão de entrada	7-12 V	7-12 V
Tensão de saída	6-20 V	6-20 V
Pinos digitais	14 (6 pinos PWM)	14 (6 pinos PWM)
Pinos analógicos	6	6
Memória Flash	16 kB	32 kB
SRAM	1 kB	2 kB
EEPROM	213 bytes	1 kB
Velocidade do relógio (Cristal Oscilador)	16 MHz	16 MHz

mesmos encontra-se no seguimento:

- Alimentação do Arduino: é o conector que permite alimentar a placa quando esta não está ligada ao computador através de um cabo USB;
- Regulador de tensão: responsável por regular a tensão;
- Entrada USB: permite a ligação a uma porta série, através de um cabo USB;
- Cristal oscilador de 16 MHz: permite o funcionamento da frequência do microcontrolador;
- Chip para comunicação com o computador: o arduino UNO utiliza o Atmega16U2 programado com um conversor USB-Serial, contrariamente a outros modelos que usam um chip FTDI USB-to-Serial;
- LEDs RX e TX: indicam os estados da comunicação com o computador;
- Botão Reset: limpa da memória o programa existente na placa Arduino, reiniciando o dispositivo;
- Pinos Digitais: são também designados I/O (Input/Output), pois funcionam como entradas e/ou como saídas digitais, utilizando comandos como *pinMode()*, *digitalWrite()* ou *digitalRead()*. Seis destes pinos digitais são designados pinos PWN (*Pulse Width Modulation*). Ainda na gama dos pinos digitais, existem dois pinos específicos, RX (*receiver*) e TX (*transmission*), que são utilizados para comunicação série com dispositivos externos, correspondendo RX à receção de dados e TX à transmissão;
- LED de alimentação: serve apenas como um sinal de estado, que indica se o Arduino está ou não a ser alimentado;
- Microcontrolador: é um circuito integrado, que contém um processador, memória e é programável, neste caso o ATmega328;
- Pinos Analógicos: utilizados para ligação a qualquer componente capaz de gerar um sinal deste tipo;
- Pinos de Alimentação: fornecem a possibilidade de alimentar dispositivos externos, como por exemplo um circuito electrónico (3,3 V e 5 V) ou moto-

res.² [42, 43]

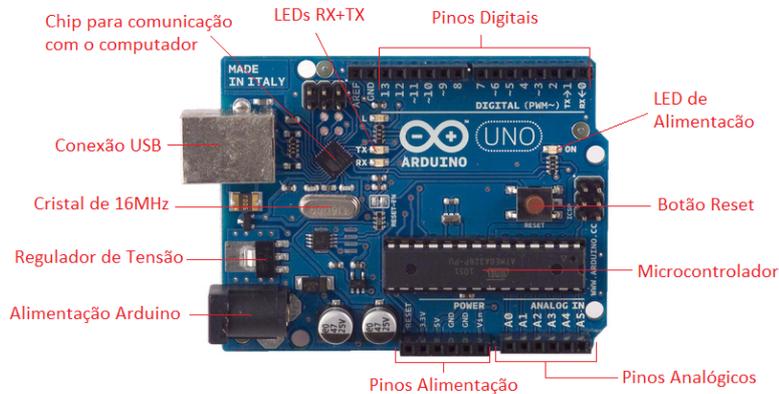


Fig. 5.2: Identificação e localização dos componentes do Arduino UNO [44].

É importante fazer uma breve descrição do microcontrolador, dada a sua importância na constituição do Arduino. Neste caso, será abordado o ATmega328. Este circuito integrado é constituído por 28 pinos, nos quais 14 são entradas e saídas digitais, 6 pinos de entrada analógica ou entrada e saída digital, 5 pinos de alimentação, 1 pino de "reset" e 2 pinos para conexão com o cristal oscilador. Através da Figura 5.3 é possível perceber de uma forma mais clara os vários pinos que constituem o microcontrolador, bem como a sua função no Arduino. É importante referir que os pins digitais 0 (RX) e 1 (TX) são responsáveis pela receção e transmissão de dados, respetivamente, no formato Série Assíncrono (USART), possibilitando a criação de uma comunicação sem-fios [45, 46].

A comunicação entre o Arduino e o computador é feita de uma forma simples, quer pelo modelo UNO, quer pelo modelo Duemilanove. O microcontrolador é responsável comunicação série e o ATmega16U2, no caso do Arduino Uno, promove o aparecimento de uma porta virtual no computador a que o Arduino está ligado. O *firmware* 16U2 utiliza os *drivers* padrão COM USB, não sendo necessário nenhum *driver* externo; excetuando quando é utilizando o Sistema Operativo *Windows*[®] em que há necessidade de um ficheiro .inf, disponível na plataforma *on-line* oficial do Arduino. Para terminar a comunicação é utilizado o ambiente de desenvolvimento integrado do Arduino (IDE), que inclui a possibilidade de visualização dos dados transmitidos através de um monitor (*Serial Monitor*). Para se obter a confirmação de que está de facto a ocorrer a transmissão de dados, os LEDs RX e TX apresentam

² <http://arduino.cc/>

um funcionamento intermitente. No caso do Arduino Duemilanove, o processo de transmissão é realizado da mesma forma, independentemente do *chip* de comunicação e do microcontrolador.

É ainda importante referir a possibilidade de diversificar as funcionalidades do Arduino através de outras placas eletrónicas com funções mais específicas - os *shields*, abordados mais à frente. Existem em grande variedade, indo desde o controlo de motores, passando por aplicações na robótica e até na implementação de sistemas sem-fios³. Para os objetivos propostos neste trabalho, na fase seguinte são utilizados *shields* para o desenvolvimento de testes de funcionalidade.

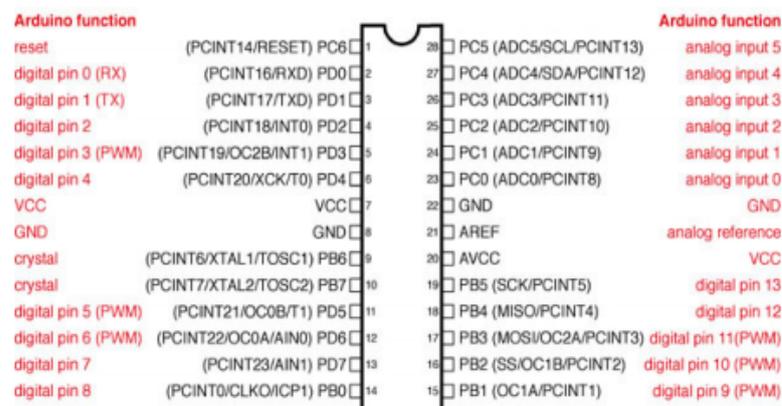


Fig. 5.3: Descrição dos pinos do microcontrolador ATmega328 (fonte: <http://computerpr0n.com/2012/04/burning-arduino-bootrom-on-an-atmega328-pu-part-2/>).

5.2 Ambientes de Desenvolvimento Integrado

Para a execução dos vários testes realizados durante este trabalho, foram utilizados diferentes ambientes de desenvolvimento integrados (IDE *Integrated Development Environment*) para a aquisição, leitura e amostragem de dados. Com o objetivo de introduzir de uma forma genérica estes programas, são nesta secção apresentadas as suas funcionalidades principais.

³ <http://shieldlist.org/>

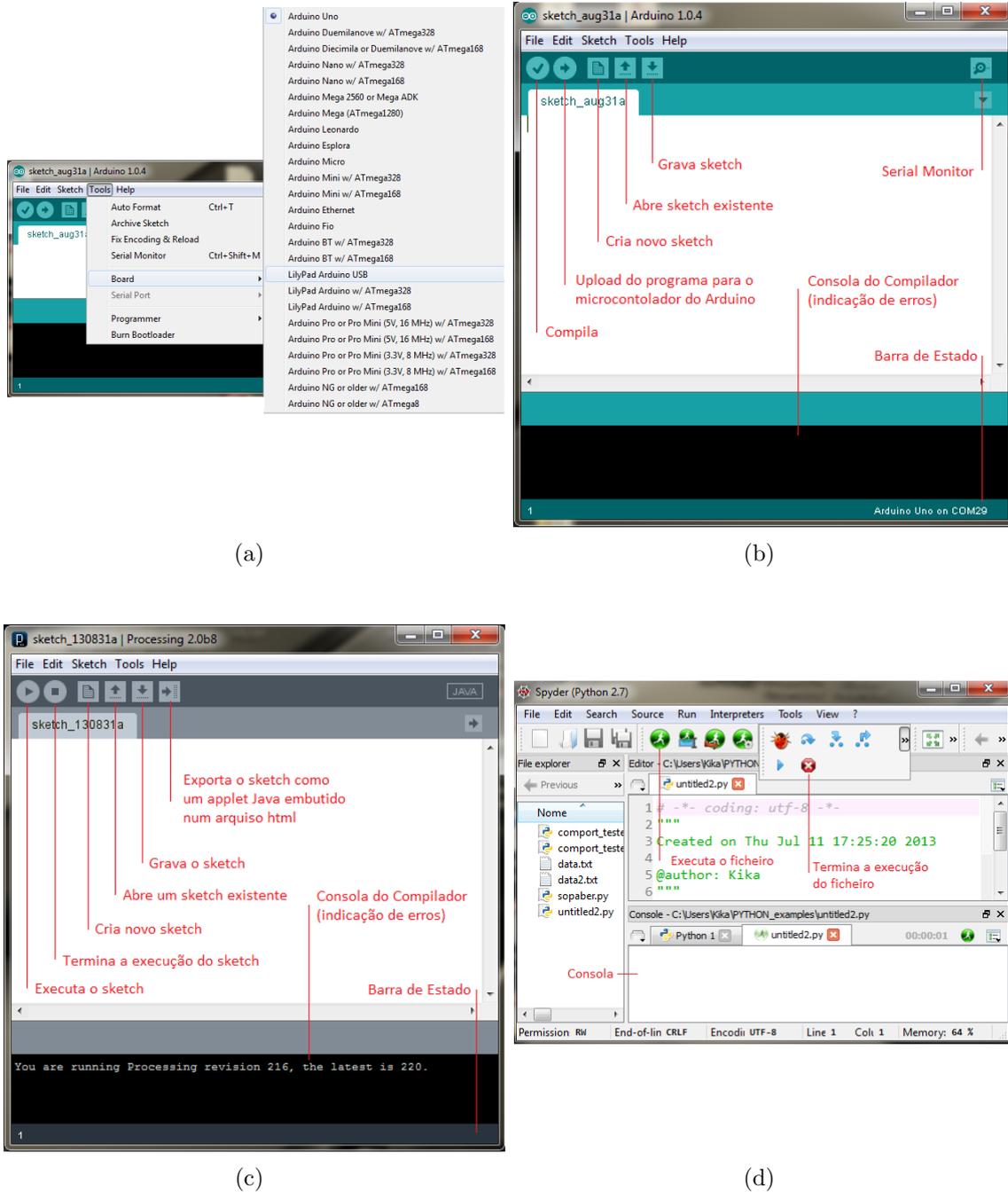


Fig. 5.4: Ambientes de desenvolvimento integrado (IDE) utilizados: (a) Arduino: menu de seleção da placa Arduino ativa; (b) Arduino (c) Processing; (d) Python.

5.2.1 Arduino

Como já foi referido anteriormente, o IDE do Arduino é um compilador que utiliza uma interface gráfica construída em Java. Este ambiente de desenvolvimento permite duas tarefas iniciais: o desenvolvimento de um programa (ou *sketch*) e o seu *upload* para o microcontrolador da placa Arduino.

Quando se inicia o ambiente de programação é essencial definir inicialmente duas opções: o tipo de placa Arduino que está ligada ao PC e a porta COM que está a ser utilizada para esse fim. Na Figura 5.4 (a), exemplifica-se este processo, onde é selecionado o modelo da placa Arduino ligada ao PC (Arduino Uno, neste caso). Após a definição da porta e da placa, é possível passar ao ambiente de desenvolvimento de programação, que é constituído por uma interface muito simples, como é possível observar na Figura 5.4 (b), onde são identificados as várias opções oferecidas pelo IDE do Arduino. É importante referir que antes de ser efetuado o *upload* de um programa para o microcontrolador do Arduino, este deve ser compilado para verificação do código através da Consola do Compilador. É ainda importante referir que este processo é automático quando se pretende fazer o *upload* dos *sketch*. O *Serial Monitor* constitui uma funcionalidade extremamente útil do IDE, visto que permite visualizar os dados provenientes de uma porta série para "debugging" dos programas [29].

5.2.2 Processing

O Processing é uma linguagem de programação simplificada baseada em Java, caracterizada por um IDE conhecido na documentação como PDE (*Processing Development Environment*). É igualmente uma ferramenta *open-source* de fácil instalação e desenvolvimento⁴ [47]. O ambiente de desenvolvimento do Processing é semelhante ao do Arduino, como se pode confirmar através da visualização da Figura 5.4 (c).

5.2.3 Python

O Python é uma linguagem de programação que permite um trabalho mais rápido e a integração dos seus sistemas de forma mais eficaz⁵. Algumas das características associadas a este tipo de linguagem é a sua sintaxe de fácil execução, linguagem simples e intuitiva, clareza, orientada a objetos (OOP) e permite o uso de extensas

⁴ <http://processing.org/>

⁵ <http://www.python.org/>

bibliotecas [48, 49, 50]. O ambiente de desenvolvimento do Python é apresentado na Figura 5.4 (d).

5.3 Testes de Funcionalidade

Esta secção diz respeito à fase de testes desenvolvidos para leitura e amostragem dos dados provenientes do circuito implementado. Inicialmente, foi necessário criar de um pequeno programa no IDE do Arduino, com o intuito de reportar os dados para o *Serial Monitor*. Numa fase posterior foi utilizado o Processing para gerar um gráfico para melhor visualização dos dados recebidos e a sua variação, em tempo real. Por fim, Python foi utilizado para visualização dos dados na sua forma definitiva. Os códigos desenvolvidos foram testados através da ligação a tensões constantes, nomeadamente a 5 V, 9 V, ligação ao GND (0 V) e, só depois de devidamente testado nas situações anteriores, ao circuito electrónico. Os testes realizados com o circuito electrónico foram desenvolvidos com o auxílio de uma fonte radioativa de 3 kBq de atividade.

5.3.1 Teste 1

Este teste, a correr no microcontrolador do Arduino, tem como objetivo a aquisição dos dados analógicos provenientes do circuito electrónico e a sua conversão para o formato digital. O código Arduino responsável pelos testes de aquisição de dados, pode ser consultado no ficheiro *readanalogdata.ino* incluído abaixo.

A função *setup()* é usada quando um *sketch* é inicializado. De seguida, através do comando *Serial.begin()*, é iniciada a comunicação série usando uma taxa de transmissão de 9600 bits por segundo (*boud rate*). Após este primeiro passo, é utilizada a função *loop()*, onde é inserido o programa do utilizador propriamente dito, neste caso com a funcionalidade de ler e registar dados, sendo executado indefinidamente, ou seja, terminados os comandos no interior da função, o processo recomeça. Para a leitura dos dados analógicos existe uma função específica designada *AnalogRead()*. Esta função é responsável pela leitura de um determinado pino analógico (0 neste caso) e retorna valores digitais entre 0 e 1023 (10 bits), que representa a tensão que passa através desse pino [51, 52]. Esta função está inserida num ciclo *for* para,

enquanto durar o ciclo, ler os valores que provêm do circuito. Por fim, é utilizado um `println()` para a escrita dos dados no *Serial Monitor*. A Figura 5.5 mostra o *Serial Monitor* onde é possível observar alguns dados adquiridos na fase de testes.

```
readanalogdata.ino

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

void loop() {
    int distsensor = 0;
    int attempts = 5;
    for (int i = 0; i < attempts ; i++) {
        distsensor += analogRead( 0 ) ;
        delay( 50 );
    }
    distsensor = distsensor / attempts ;
    Serial.println( distsensor );
}
```

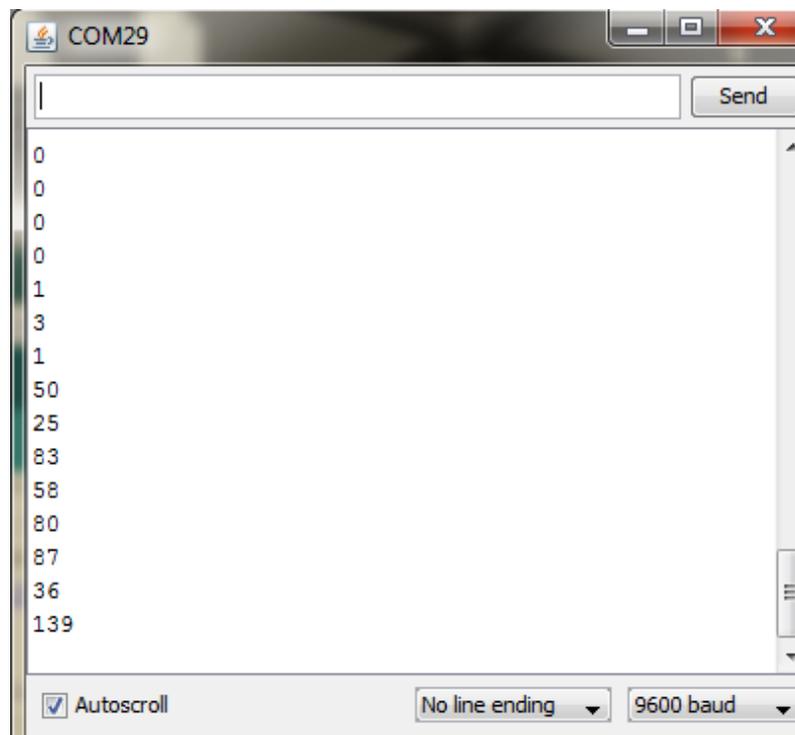


Fig. 5.5: *Serial Monitor* na fase de testes onde estão representados os dados provenientes do circuito eletrônico, ligado ao tubo GM.

Inicialmente não existe qualquer fonte radioativa perto sensor (tubo GM), o que

resulta 0, com uma margem de erro de 5 valores. Após a aproximação da fonte ao sensor, é claramente visível a alteração dos dados para valores bastante superiores. É importante lembrar que os dados provenientes do circuito, e em resultado da aproximação da fonte radioativa, têm origem num processo de decaimento radioativo como já foi referido anteriormente. Como tal, é expectável que estes valores não sejam contínuos e flutuem dentro de uma determinada gama de valores, consoante a deteção obtida pelo sensor e atividade da fonte. Desta modo, foi possível confirmar a deteção da radiação ionizante no tubo, não apenas através de um sistema acústico.

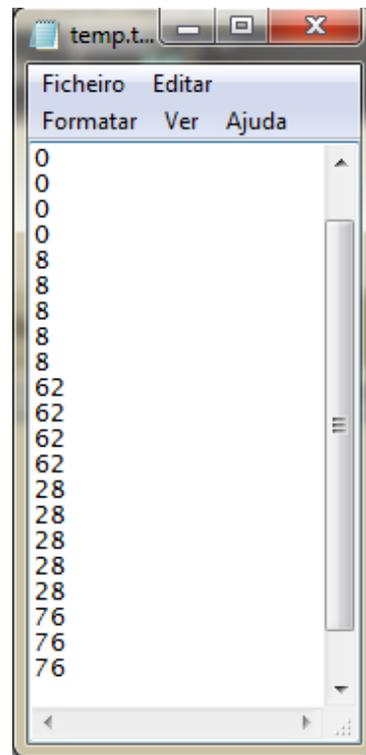
5.3.2 Teste 2

Utilizando o mesmo programa no Arduino apresentado no Teste 1, foi implementado um código em Processing com a funcionalidade de ler os dados digitalizados provenientes do circuito e representá-los num gráfico (ficheiro *detetor.pde* presente no Apêndice A).

O programa inicia com a importação de uma biblioteca, *processing.serial*, utilizada especificamente para a comunicação série. De seguida, além da criação do ambiente gráfico, foi realizado o tratamento dos dados que chegam à porta série e consequente gravação para um ficheiro no formato .txt. É então criado um *buffer*, ou seja, uma *string* vazia e uma variável NEWLINE definida como 10, que no código Ascii (apresentado no Apêndice B) corresponde a `"/n"` (mudança de linha). Um ciclo *while* é utilizado para garantir que os comandos são apenas executados quando se dá um evento na porta série ou, por outras palavras, enquanto não forem detetados eventos, a função *serialEvent* não é inicializada. É importante lembrar que os dados que entram na porta série chegam já num formato específico, ou seja, o número seguido de uma mudança de linha, `"/n"`, que não é obviamente visível quando os dados são lidos no *Serial Monitor* apresentado anteriormente. Deste modo, o código foi desenhado de modo a procurar o carácter `"/n"`, visto indicar o final da *string* que chega à porta série. A cada `"/n"` encontrado, os caracteres anteriores são incluídos no *buffer* e é removido o carácter `"/n"` que já não é necessário pois só interessa o valor e este é convertido para um número inteiro através da função *Integer.parseInt*. Neste caso, dividiu-se por 2 o resultado anterior de modo a obter-se uma escala mais reduzida de valores. Para finalizar, a *string* agora preenchida deverá ficar vazia para o processo se iniciar novamente.



(a)



(b)

Fig. 5.6: Resultados do teste 2: (a) Representação gráfica dos dados provenientes do circuito com recurso ao Processing; (b) Excerto do ficheiro no formato .txt, contendo os dados provenientes do circuito de deteção.

Após o tratamento dos dados, é necessário proceder à sua gravação num ficheiro. Para tal, um ficheiro vazio no formato .txt é criado com o nome desejado, através do comando *createWriter()*, que é depois preenchido com os números inteiros provenientes do processo anterior.

Pela análise do gráfico da Figura 5.6 (a) é possível verificar a variação do sinal quando o tubo GM se encontra exposto à fonte radioativa. Inicialmente, a fonte está afastada do tubo GM e, após uma aproximação, é visível a alteração dos valores. Mais uma vez, o gráfico corrobora o comportamento associado a este tipo de sinal, dada a grande oscilação que apresenta. Na Figura 5.6 (b) é possível observar um excerto dos dados correspondentes ao gráfico.

5.3.3 Teste 3

Neste teste foi implementado um código com algumas diferenças relativamente ao teste anterior. Contudo, a funcionalidade manteve-se no essencial, como é possível observar no código apresentado abaixo (*readanalogdata2.ino*). É definida a porta analógica de uma outra forma e a diferença mais substancial consiste na utilização de uma função de mapeamento dos valores. A função *map()* é definida por 5 parâmetros: *value*, *fromLow*, *fromHigh*, *toLow* e *toHigh*, nesta ordem. Desta forma, dado um determinado '*value*', neste caso o sinal proveniente da porta analógica 0, a função mapeia de um intervalo inicial (0 – *fromLow* a 1023 – *fromHigh*) para um novo intervalo de valores (0 – *toLow* a 255 – *toHigh*)⁶. Por fim, é executado um *delay()*, neste caso de 1000 milissegundos. Este programa foi testado da mesma forma que os anteriores e os dados foram apresentados no *Serial Monitor* do ambiente de desenvolvimento do Arduino.

readanalogdata2.ino

```
int potPin = 0;

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

void loop()
{
  int val = map(analogRead(potPin), 0, 1023, 0, 255);
  Serial.println(val);
  delay(1000);
}
```

Código desenvolvido em Python será posteriormente utilizado para ler os dados que chegam à porta série (ficheiro *serialread.py*, apresentado abaixo). Pretende-se com este código a visualização dos dados em tempo real, a ativação de um alarme (caso os valores sejam superiores a um valor estipulado) e gravação dos dados para um ficheiro, quando o alarme é ativado. O programa inicia com a importação de duas bibliotecas, uma relativa à comunicação série e outra utilizada para aceder à hora e data do sistema. Posteriormente, é indicada a porta COM a usar, uma taxa de transmissão de 9600 bits por segundo e, finalmente, aberta a porta série para permitir o acesso aos dados. Antes de dar início ao processo de aquisição de dados

⁶ <http://arduino.cc/>

pela porta série, é feito um teste para verificar se a mesma se encontra aberta. Desta forma, enquanto a porta estiver aberta, o programa está permanentemente a aceder à porta série para adquirir valores. Caso esteja fechada, é executado o comando para a abrir, recorrendo à estrutura de controlo *if* e *else* [49, 53]. Estes valores, juntamente com a hora do sistema, são colocados numa *string* que é depois inserida num ficheiro de texto. Relativamente ao alarme, caso o valor recebido seja superior a 4, é gerado um alarme e os valores são gravados para um ficheiro. É importante referir que o valor 4 foi escolhido, com base nos testes realizados, nos quais foi possível concluir que quando não existe radiação nas proximidades do tubo, o valor mantém-se 0, com uma margem de erro de cerca de 5 valores. A Figura 5.7 (b), por seu lado, apresenta o ficheiro criado com a hora e valores lidos aquando da ativação do alarme.

serialread.py

```
import serial
import datetime

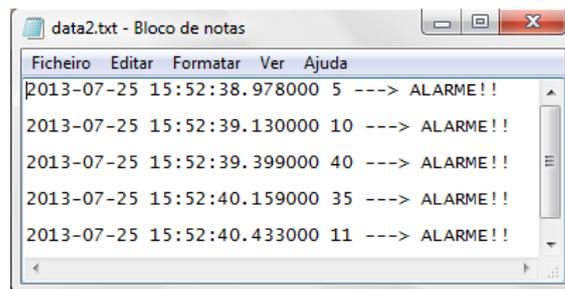
ser = serial.Serial() # open first serial port
ser.port = 28
ser.baudrate = 9600
ser.open()

print ser.portstr + "_is_open?" + str(ser.isOpen())

f = open("data2.txt", "w")

while (1):
    if ser.isOpen():
        t = datetime.datetime.now()
        val = int( ser.read(3) )
        if val > 4:
            s = ("%s %g -> %s\n") % (str(t), val, "ALARMEEE!!")
            f.write(s)
            print s
        else:
            print str(t), str(val)
    else:
        ser.open()

f.close()
```



```
data2.txt - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
2013-07-25 15:52:38.978000 5 ---> ALARME!!
2013-07-25 15:52:39.130000 10 ---> ALARME!!
2013-07-25 15:52:39.399000 40 ---> ALARME!!
2013-07-25 15:52:40.159000 35 ---> ALARME!!
2013-07-25 15:52:40.433000 11 ---> ALARME!!
```

(a)

Fig. 5.7: Resultados do teste 3: Ficheiro de texto gerado pelo Python quando o alarme é ativado.

Redes Sem-Fios

No capítulo 6, pretende mostrar-se a forma como foram efetuados os testes de comunicações sem-fios, baseados no protocolo *ZigBee*. Uma rede de sensores sem-fios (RSSF) consiste num *array* de dispositivos sem-fios, mais especificamente sensores, distribuídos numa determinada região que se pretende monitorizar. Tal obrigou a um estudo prévio de vários conceitos associados às redes informáticas. Inicialmente, foi abordado o protocolo *ZigBee*, com especificação das camadas que o constituem, tipo de nós, dispositivos que a formam e topologias da rede. De seguida, foram estudados os módulos XBee que serão utilizados para a construção da rede, os vários tipos de antenas existentes, *Shields*, modos de operação e descrição dos adaptadores. Após esta abordagem, foi possível passar à fase de testes, com o objetivo de verificar a funcionalidade dos dispositivos, apoiados numa rede de sensores sem-fios, com base no protocolo *ZigBee*. Cada sensor da rede corresponde a um circuito eletrónico com um tubo GM associado, que transmite via *wireless* os dados relativos à deteção de radiação ionizante.

6.1 ZigBee

Os termos ZigBee e XBee são muitas vezes confundidos, mas estes dois são distintos. ZigBee é um protocolo de comunicação sem-fios com baixo consumo de energia e sem-fios, enquanto que o XBee é um dispositivo de rádio frequência (RF), que suporta uma variedade de protocolos de comunicação, nomeadamente o ZigBee, 802.15.4, *Wi-Fi*, entre outros. O protocolo ZigBee foi desenvolvido pela *ZigBee*

*Alliance*¹, permitindo a implementação de sistemas que necessitem de um baixo consumo de energia, para transferências de dados e custo reduzidos [54, 55]. O padrão ZigBee foi desenvolvido com base no protocolo IEEE 802.15.4, patenteado pelo IEEE, utilizado para redes de velocidade e consumo de energia reduzidos. Este protocolo opera numa frequência de 2400 MHz a 2483,5 MHz (modulação) e taxa de transmissão ou transferência de dados de 250 kbps [56].

6.1.1 Modelo de Camadas

Recordando o modelo OSI (*Open Systems Interconnection*), este agrupa as funcionalidades de comunicação em sete camadas distintas consoante os critérios de afinidade. São estas as camadas física, ligação de dados, rede, transporte, sessão, apresentação e aplicação. Com base no modelo OSI, são aqui apresentadas as diferentes camadas para o padrão ZigBee, baseado no protocolo IEEE 802.15.4 (Figura 6.1) e especificadas abaixo. É importante referir que as duas primeiras camadas são definidas pelo protocolo IEEE 802.15.4, enquanto que as restantes são definidas pelo protocolo ZigBee.

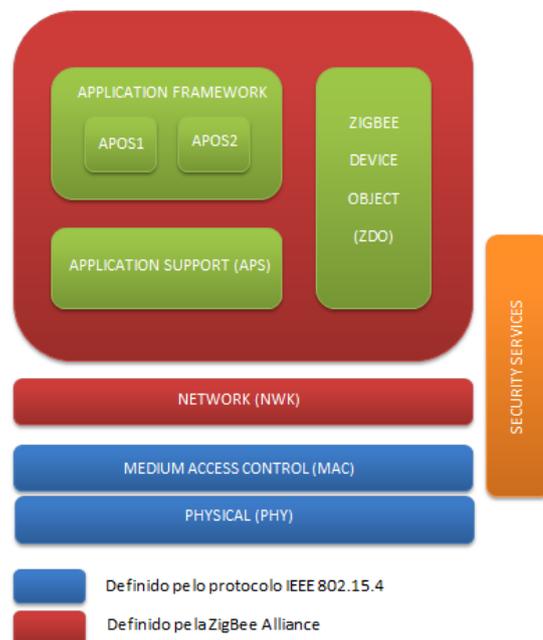


Fig. 6.1: Camadas do padrão ZigBee.

¹ <http://www.zigbee.org/>

Camada Física (PHY – *Physical Layer*)

Tendo em conta o modelo OSI, esta camada tem como funcionalidade fazer a interface com o meio físico, isto é, transmissão de dados através de um canal de comunicação. É necessário para isso a definição de parâmetros como a modulação e taxa de transferência, já mencionados anteriormente para o protocolo IEEE 802.15.4. Utiliza a técnica de Transmissão de Sequência Direta (DSSSF – *Direct Sequence Spread Spectrum*), o que permite o uso de equipamentos simples e de baixo custo [57, 58].

Ligação de dados (MAC – *Media Access Control*)

Nesta camada é realizada a gestão da transmissão dos dados entre vários nós e gestão de acesso aos canais. É responsável pela sincronização de *beacons*, validação de dispositivos presentes na rede, associação e dissociação destes. Inclui o serviço de reenvio de transmissões e de técnicas de prevenção de colisão *Carrier Sense Multiple Access with Collision Avoidance* [59, 60, 61]. Existem dois métodos de acesso a canais, isto é, dois modos de operação da rede: *beaconing* e *non-beaconing*.

- *Beaconing*: os *routers* que fazem parte da rede transmitem num intervalo sincronizado de tempo um sinal para confirmação da presença dos outros dispositivos, estes dados enviados são designados *beacons*. Assim, estes dispositivos permanecem no modo *sleep*, sendo ativados apenas quando necessário. Utiliza o mecanismo CSMA-CA. O *beaconing* promove um custo energético bastante baixo [62].
- *Non-beaconing*: é necessário que os dispositivos estejam sempre ativos, o que promove um consumo de energia mais elevado, sendo, no entanto um processo mais flexível. Por outro lado, não é necessário o envio de *beacons* em intervalos sincronizados de tempo. Não utiliza o mecanismo CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*) [62, 54].

Rede (NWK – *Network*)

Esta camada serve como suporte à camada de aplicação e executa o transporte de dados. Para o padrão *ZigBee* esta camada foi concebida com o intuito de criar redes de grandes dimensões e sem necessidade de dispositivos com potências elevadas. Tem como principais responsabilidades detetar canais ativos na comunicação (*Network Scan*), criar uma nova rede ou juntar-se a uma já existente (*Creating/Joining a*

PAN), descobrir dispositivos no canal ativo da PAN - *Personal Area Network* - (*Device Discovery*) e definir a topologia de rede pretendida [63, 64].

Aplicação (*Application Layer*)

É composta por um conjunto de *Applications Objects* distribuídos pelos vários nós da rede. Pode ser dividida em três sub-camadas: *Application Support* (APS), *Application Objects* (APOs) e *ZigBee Device Object*(ZDO) [63, 64].

- *Application Support* (APS): é considerada uma interface entre a camada de rede e a camada da aplicação e encaminha os dados provenientes da camada de rede para diferentes dispositivos que estão em funcionamento nos nós, mantendo uma espécie de base de dados com os vários dispositivos que constituem a rede;
- *Application Objects* (APOs): é o *software* que permite controlar uma unidade de *hardware*, ou seja, atribui funções aos vários dispositivos. Estas funções contidas pelos objetos são definidas pelos fabricantes, em que cada objeto representa um tipo/perfil de aplicação diferente (consegue armazenar 240 objetos) [65];
- *ZigBee Device Object* (ZDO): responsável por implementar o perfil do dispositivo, implementando as funcionalidades básicas para permitir a comunicação entre a APS e os APOs. Além disto, é responsável pela procura e identificação de novos dispositivos na rede e os serviços que oferecem, tais como os serviços de localização do dispositivo e definição de configurações de segurança da rede [63, 66];

Serviços de Segurança (*Security Services*)

Inclui as camadas MAC, NWK e APS e possui a função de implementar esquemas de segurança em cada uma destas camadas, através de chaves de segurança e comunicações seguras. Implementa o padrão AES (*Advanced Encryption Standard*) de 128 bits. O ZDO é responsável por definir as políticas e configurações de segurança, que são depois implementadas pela camada *Security Services* [63].

6.1.2 Tipos de Nós

Uma rede é constituída por nós, podendo estes ser distinguidos em dois tipos: *Full-function device* (FFD) e *Reduced-Function Devices* (RFD).

- *Full-function device*: podem ser *Coordinators*, *Routers* ou *End Points* e podem comunicar com qualquer dispositivo da rede. Neste caso, a implementação mais complexa.
- *Reduced-Function Devices*: dispositivos de implementação mais simples e limitados à configuração da topologia em estrela. Pode apenas comunicar com FFDs e nunca são *Coordinators* numa PAN [57, 67].

6.1.3 Dispositivos numa rede *ZigBee*

Uma rede *ZigBee* tem de ter obrigatoriamente pelo menos um *Coordinator* e pelo menos um outro dispositivo, sendo este um *Router* ou um *End Device*. Na Tabela 6.1 é feita uma breve descrição das funcionalidades associadas a estes dispositivos na camada de rede.

Tab. 6.1: Funções dos ZC, ZR e ZED na camada de rede [68, 64].

Função na camada de rede	ZC	ZR	ZED
Cria uma rede <i>ZigBee</i>	x		
Atribui endereços à rede	x	x	
Gerir a associação e dissociação de outros dispositivos	x	x	
Contém uma lista de dispositivos vizinhos	x	x	
Permite a passagem de pacotes da camada de rede	x	x	
Transmite pacotes da camada de rede	x	x	x
Associação e dissociação à rede <i>ZigBee</i>	x	x	x

- *ZigBee Coordinator* (ZC): também designado como PAN *Coordinator*, é responsável por criar a rede, gerar a sua identificação (PAN ID), distribuir endereços e gerir a rede de segurança. Numa rede *ZigBee* apenas pode existir um único ZC [68];
- *ZigBee Router* (ZR): o *Router* funciona como um mensageiro, que tem a funcionalidade de enviar e receber dados, ou seja, pode passar dados provenientes de dispositivos para outros através dele e pode associar-se a redes já existentes. Uma rede pode ser formada por vários ZR;

- *ZigBee End Device* (ZED): os *End Device* juntam-se a redes já existentes e recebem e enviam informações, mas apenas sobre o próprio dispositivo, isto é, não agem como ZRs que passam a informação proveniente de outros dispositivos. Como tal, os ZED necessitam sempre de um dispositivo *Master*, ZC ou um ZR, para se ligar à rede. Estes dispositivos possuem a característica de permanecerem no modo *Sleep* quando não estão a ser utilizados, sendo o *Master* responsável por armazenar a informação enquanto o ZED está neste modo, contribuindo desta forma para a poupança de energia do dispositivo [69].

6.1.4 Topologias de Rede

Os vários dispositivos que formam as redes ZigBee podem ser posicionados de diferentes formas, criando assim distintas topologias de rede: *Per* (Par), *Mesh* (Malha), *Cluster Tree* (Árvore) e *Star* (Estrela).

Per

Esta é a topologia de rede mais simples em que são apenas utilizados dois dispositivos, o que leva à existência de dois nós. Um dos nós é obrigatoriamente constituído pelo *Coordinator* (FFD) e o outro nó poderá ser um *Router* (FFD) ou um *End Device* (FFD ou RFD). É possível visualizar estas duas configurações na Figura 6.2 [69].



Fig. 6.2: Topologia *Per*.

Star

A configuração desta topologia de rede também é simples, com o *Coordinator* (FFD) a ocupar uma posição central na rede, ficando ligado a um círculo de *End Devices* (FFD ou RFD). Desta forma, todas as mensagens que estão no sistema passam obrigatoriamente pelo *Coordinator*, que as encaminha quando necessário para o dispositivo pretendido, já que os *End Devices* não conseguem comunicar diretamente

entre si. Se uma das ligações for interrompida ou falhar, apenas o nó referente a essa ligação é afetado. No entanto, se o *Coordinator* deixar de funcionar, toda a rede é destruída (Figura 6.3) [70].

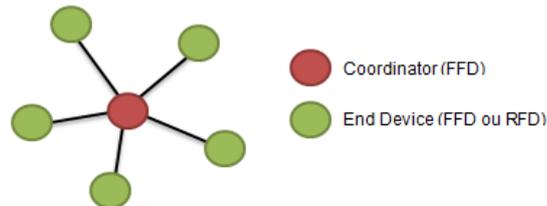


Fig. 6.3: Topologia *Star*.

Mesh

A topologia *Mesh* é um pouco mais complexa que as anteriores. Inclui nós constituídos por *Routers* (FFD), além do obrigatório *Coordinator* (FFD). Este apenas gere a rede e funciona da mesma forma que os *Routers*, encaminhando as mensagens provenientes dos outros dispositivos. Os *End Devices* (FFD ou RFD), como foi referido anteriormente, não podem comunicar entre si, utilizando os *Routers* e *Coordinator* para passar a informação. Tendo em conta que existem vários caminhos possíveis para os diversos nós, esta configuração torna-se mais robusta no que diz respeito a tolerância a falhas, que possam ocorrer em nós ou ligações individuais (Figura 6.4) [70, 61, 64].

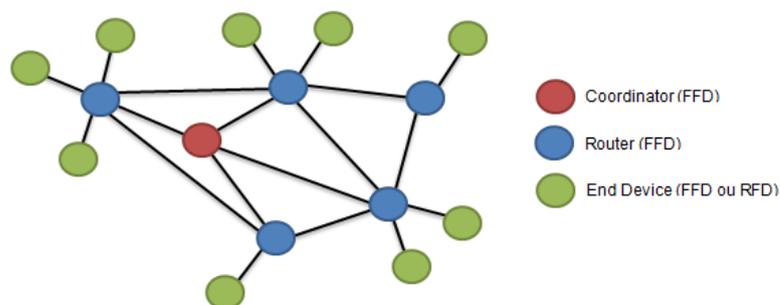


Fig. 6.4: Topologia *Mesh*.

Cluster Tree

De um modo geral, o *Coordinator* (FFD) juntamente com os *Routers* (FFD) existentes na rede formam uma espécie de ramificação, constituindo o conjunto de ramificações um *Cluster*. O PAN *Coordinator* é nomeado para identificar a rede e é sincronizado com os restantes dispositivos. Os RFD são então ligados a um *Cluster* no final da ramificação, através da conexão a um FFD. O primeiro *Cluster* criado, é obviamente formado pelo *Coordinator* da rede e designado como *Cluster Head* (CLH), atribuindo um valor ao *Cluster*, designado *Cluster Identifier* (CID), igual a zero. O *Coordinator* escolhe agora um identificador para a PAN que não esteja em uso e envia *beacons* para os dispositivos vizinhos, que recebem a informação e questionam o CLH para se juntar à PAN criada. Para cada *Cluster* que se associa a rede, é necessário um CLH, sendo atribuído o CID por ordem crescente [70]. No exemplo da Figura 6.5 é possível observar os *Clusters* representados a laranja, o seu CID e o CLH associado a cada *Cluster* [41].

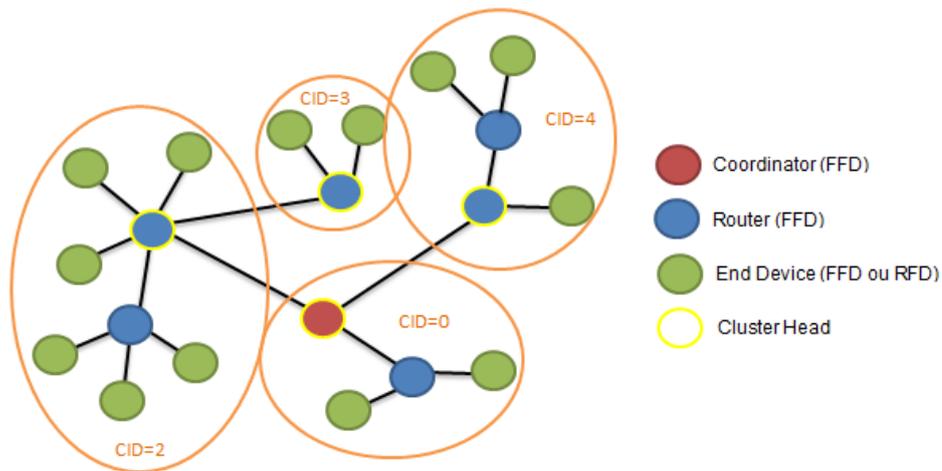


Fig. 6.5: Topologia *Cluster Tree*.

6.2 Módulos XBee

No âmbito do presente estudo foram utilizados módulo XBee que correspondem à variante série 2, pois além de mais económicos, possuem características suficientes para o que se pretende realizar neste trabalho [69].

6.2.1 Pinos dos Módulos XBee

Estes módulos estão organizados em 20 pinos de funcionalidades específicas. Na Figura 6.6 pode ser observada uma representação esquemática do módulo com identificação de cada um dos pinos que o compõem e na Tabela 6.2 a descrição e função dos mesmos.

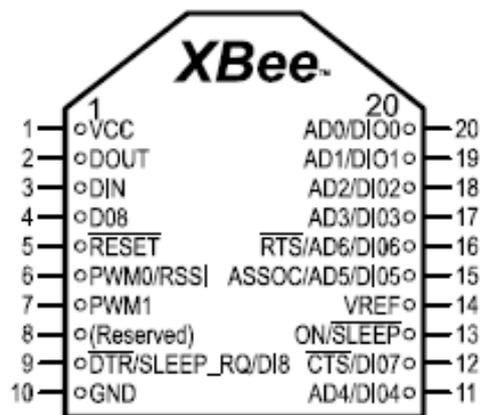


Fig. 6.6: Pinos do módulo XBee [30].

6.2.2 Shield XBee

O *Shield* XBee permite a comunicação entre o módulo XBee e a placa Arduino, que possui pinos específicos para a adição do módulo XBee. Para o desenvolvimento de alguns dos testes realizados foi utilizado o *Shield Wireless* SD, ilustrado na Figura 6.7 (a). É necessário ter em conta que este *Shield* possui uma forma de determinar como a comunicação série do módulo XBee se liga à comunicação série entre o microcontrolador e o chip USB-to-Serial: posição Micro ou USB. Quando o *Shield* está no modo USB (Figura 6.7 (b)), o pino DOUT no módulo XBee é ligado ao pino RX do conversor USB-to-Serial e o pino DIN do módulo ao pino TX do conversor USB-to-Serial. Esta configuração permite que o módulo comunique diretamente com o computador, ou seja, ao microcontrolador na placa Arduino é feito um *bypass*. Para utilização do *Shield* neste modo, o microcontrolador deverá ser retirado da placa Arduino ou programar o microcontrolador com um programa vazio (`void setup()` `void loop()`). É ainda possível fazer a ligação do pino GND ao pino RESET. Desta forma, é possível configurar o módulo XBee através do *software*

Tab. 6.2: Descrição dos pinos do módulo Xbee [71, 30, 59].

Nº	Nome	Descrição
1	VCC	Alimentação (3,3V).
2	DOUT	Saída (DOUT) de dados. Ligado normalmente ao microcontrolador para enviar e receber dados de um sensor.
3	DIN	Entrada (DIN) de dados. Ligado normalmente ao microcontrolador para enviar e receber dados de um sensor.
4	DIO12	Entrada ou saída digital.
5	RESET	A função de RESET é iniciada quando o pin 5 é conectado ao pino 10 (GND).
6	RSSI PWM/DIO10	Indicador da força do sinal recebido (RX). Entrada ou saída digital.
7	DIO11	Entrada ou saída digital.
8	RESERVED	Não conectar.
9	Sleep/DIO8	Usado para ativar ou desativar o modo <i>Sleep</i> do módulo. Quando está no nível alto (aplicação de tensão) o módulo fica no modo <i>sleep</i> , quando está no nível baixo (sem aplicação de tensão ou ligado ao GND) o módulo volta a sua funcionalidade. Entrada ou saída digital.
10	GND	Terra.
11	DIO4	Entrada ou saída digital.
12	CTS/DIO7	Controlo de fluxo <i>Clear-to-Send</i> (CTS). Entrada ou saída digital.
13	ON/SLEEP	Indicativo do estado <i>sleep</i> do módulo, através da utilização de um LED: quando no estado aceso indica que módulo está a operar normalmente e quando desligado indica que o módulo se encontra no modo <i>sleep</i> .
14	VREF	Tensão de referência
15	ASSOC/DIO5	Dá a indicação se o módulo está conectado a uma rede através da utilização de um LED: quando o módulo não está conectado a uma rede, o LED está ligado e quando está conectado pisca em intervalos regulares. Entrada ou saída digital.
16	RTS/DIO6	Controlo de fluxo <i>Request-to-Send</i> (RST). Entrada ou saída digital.
17	AD3/DIO3	Entrada analógica. Entrada ou saída digital.
18	AD2/DIO2	
19	AD1/DIO1	
20	AD0/DIO0	

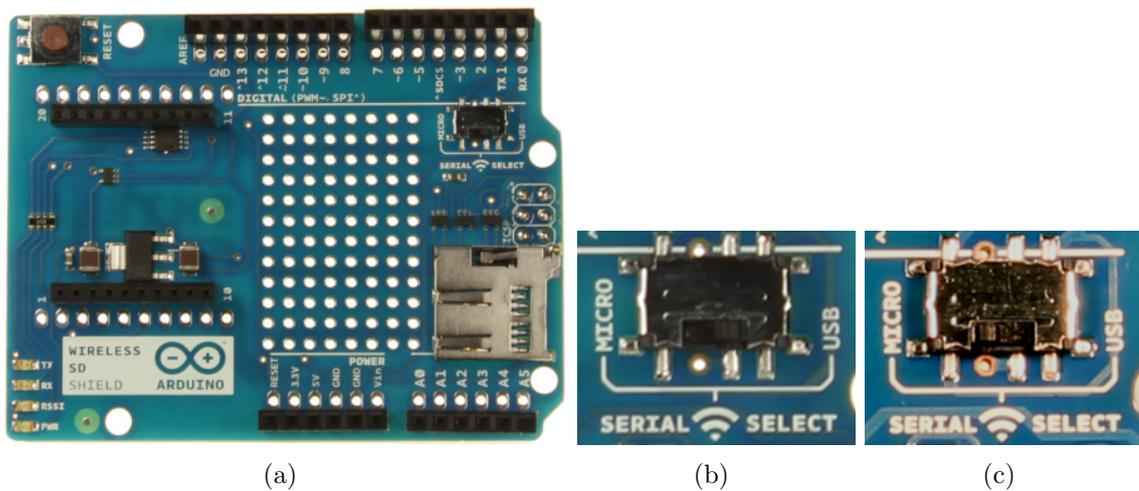


Fig. 6.7: *Shield Xbee Wireless SD*: (a) *Shield*; (b) Modo USB; (c) Modo Micro.

X-CTU, abordado mais a frente. Quando o *Shield* está ajustado para a posição Micro, Figura 6.7 (c), o pino DOUT do módulo XBee é ligado ao pino RX do microcontrolador e o pino DIN ao pino TX. Desta forma, o módulo passa a estar em comunicação com o microcontrolador do Arduino. De notar que, neste modo, o microcontrolador não é programável².

6.2.3 Antenas

Para transmitir e receber sinais, estes dispositivos necessitam de antenas, que podem ser de vários tipos. Um dos aspetos em que diferem as antenas é o ganho a estas associado, isto é, a energia de transmissão ou padrão de transmissão³. São aqui apresentados as mais relevantes:

- *Wire* ou *Whip*: consiste num fio perpendicular ao dispositivo com 2,5 cm de comprimento e a sua transmissão é omnidireccional, isto é, a distância de transmissão máxima é idêntica em qualquer direção quando o fio se encontra na vertical (Figura 6.8 (a));
- *Chip*: trata-se de uma placa em cerâmica, plana e alinhada com o módulo XBee. Este tipo de antena torna o dispositivo mais compacto e resistente, tornando-o adequado para uso em locais de dimensão reduzida ou quando a antena possa estar sujeita a esforços que poderiam danificá-la, caso fosse

² <http://arduino.cc/en/Main/ArduinoWirelessShield>

³ <http://www.digi.com/lp/xbee/>

utilizado por exemplo, uma antena do tipo *Wire*. Uma das desvantagens deste tipo de antena é o facto do seu padrão de transmissão ser cardióide (em forma de coração), isto é, o sinal é atenuado em certas direções (Figura 6.8 (b));

- Conector U.FL: Este conector é utilizado em situações muito específicas. Quando é necessário obter um sinal do interior de uma caixa metálica, por exemplo, é necessário que a antena, responsável pela transmissão dos dados, fique no exterior da caixa, pois no seu interior, o sinal seria atenuado pela caixa. Assim, surge este tipo de solução, que permite a ligação de uma antena específica para uma determinada situação, como por exemplo uma orientação específica da antena, antenas com ganhos elevados ou alcances mais longos, etc. Este conector é de tamanho reduzido, relativamente frágil e ligado a um fio que, por sua vez, se liga a uma antena externa (Figura 6.8 (c));
- Conector RPSMA: Este conector também permite a ligação a uma antena externa com características mais específicas. É maior que o conector referido anteriormente, possibilitando, contudo, fazer a ligação de uma antena externa diretamente ao conector, sem o uso de um fio de ligação (Figura 6.8 (d)) [69].

Para a implementação do *array* de sensores, foram utilizados os módulos XBee do tipo *wire* e *chip*.

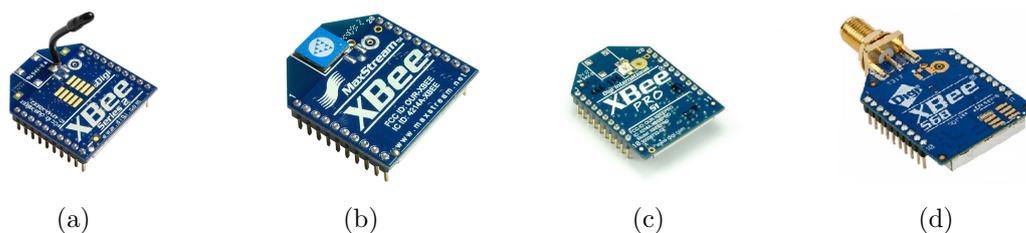


Fig. 6.8: Tipos de antenas dos módulos XBee: (a) *Wire* ou *Whip*; (b) *Chip*; (c) Conector U.FL; (d) Conector RPSMA.

6.2.4 Adaptadores

Para configurar cada módulo XBee de forma mais prática e sem haver a necessidade de remover o microcontrolador da placa Arduino, optou-se pelo uso de adaptadores, que permitem a ligação do módulo ao PC via USB. Existem vários tipos de adaptadores, optando-se, porém, por apenas um tipo de adaptadores: o

Adafruit XBee Adapter⁴. Este adaptador, apesar de ser de baixo custo, necessita de um acessório adicional, um cabo FTDI que permite a ligação à porta USB do PC. É possível observar na Figura 6.9 o adaptador conectado ao módulo XBee e ao cabo FDTI.



Fig. 6.9: Adaptador com o módulo XBee e cabo FTDI conectados.

Soldadura

Estes componentes necessitam ser devidamente soldados à placa, já que são comercializados em separado. Os vários componentes necessários à montagem do adaptador, encontram-se descritos na Tabela 6.3⁵.

Na Figura 6.10, são apresentados os vários passos a realizar para soldar de cada componente. Os melhores resultados são obtidos com recurso a um suporte, por forma a estabilizar a placa impressa. Após o posicionamento de cada componente no devido local, os seus terminais são ligeiramente dobrados para garantir que estes não se movam (Figura 6.10 (a)). Utilizando um ferro e fio de solda, o componente fica soldado após aproximadamente 2 segundos (Figura 6.10 (b) e (c)). Para finalizar o processo, corta-se os terminais salientes do componente com um alicate (Figura 6.10 (d)).

Os primeiros componentes soldados foram as resistências, na ordem R3, R2 e R1. Como não possuem polaridade, não é relevante a posição em que são colocadas. As resistências R1 e R2 são responsáveis por ajustar a intensidade de luz nos LEDs que irão compor a placa. Depois das resistências é feita a solda dos LEDs, que possuindo polaridade, obriga a verificar claramente as indicações desenhadas na placa para que o terminal de maior comprimento do LED seja colocado no polo positivo (+) e o mais curto no negativo (-). O LED vermelho, ASC, indica se o XBee está a funcionar corretamente e o LED verde, RSSI (*Received Signal Strength Indication*), indica

⁴ <http://www.ladyada.net/make/xbee/index.html>

⁵ <http://www.ladyada.net/make/xbee/parts.html>

Tab. 6.3: Lista de Componentes que compõem o adaptador XBee da Adafruit

Imagem	Nome	Descrição	Quantidade
	IC1	Regulador de tensão de 3.3 V	1
	IC2	Circuito integrado	1
	LED1	LED vermelho	1
	LED2	LED verde	1
	R1,R2	Resistência 1 k Ω	2
	R3	Resistência 10 k Ω	1
	C1	Condensador 0,1 μ F	1
	C2	Condensador 47 μ F/4 V	1
		Pente macho	1
		Pente fêmea	1
	PCB	Placa impressa	1

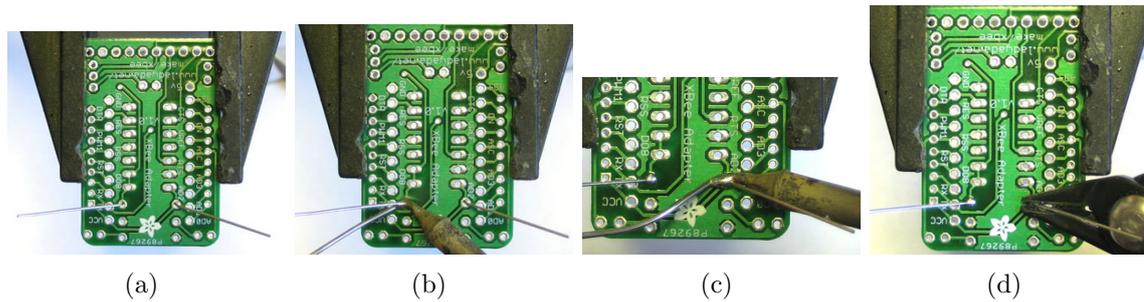


Fig. 6.10: Procedimento para a soldadura dos componentes: (a) dobrar os terminais do componente; (b) aproximar o ferro e o fio de solda; (c) esperar 2 segundos para fundir a solda; (d) cortar os terminais salientes do componente.

quando os dados estão a ser recebidos pela unidade. Os componentes a soldar de seguida são os condensadores C1 e C2. O condensador C1 não possui polaridade, sendo um condensador de cerâmica, com a função de filtrar o ruído de alta frequência proveniente da fonte de alimentação. Em contrapartida, o condensador C2, do tipo eletrolítico, possui polaridade. Este condensador é responsável por manter a fonte de alimentação de 3,3 V estável. De seguida, é soldado o transístor (MCP1700-33), que deverá ser colocado na posição correta, consoante a forma como está desenhada na placa. Funciona nesta situação como regulador, transformando a entrada de 5 V nos 3,3 V que alimenta o XBee. Posteriormente, foi soldado o CI, que possui cinco circuitos lógicos para ligar ao XBee. Deverá ser identificado o entalhe em forma de 'u' presente no próprio *chip* e na placa, para a correta orientação do CI. Devido à proximidade dos pinos, a quantidade de solda requerida para esta situação é menor, para que não haja ligação entre eles. A soldadura dos pentes fêmea deverá ser realizada preferencialmente por duas pessoas, para um correto alinhamento dos mesmos. Por fim, para a solda do pente, é necessário parti-lo com a ajuda de um alicate, tendo em conta o número necessário de pinos para o CI. O adaptador XBee com todos os componentes soldados pode ser observado na Figura 6.11.

Os pinos do adaptador que permitem a ligação ao cabo FTDI e, posteriormente, ao PC, são apresentados e descritos na Figura 6.12. O cabo FDTI permite a comunicação de dados do adaptador com o PC.

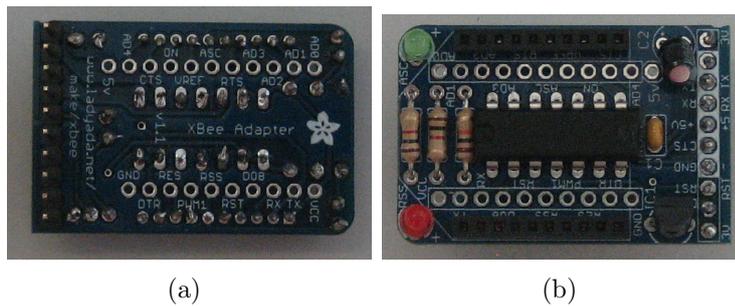


Fig. 6.11: Adaptador com os componentes soldados: (a) vista de trás; (b) vista de frente.

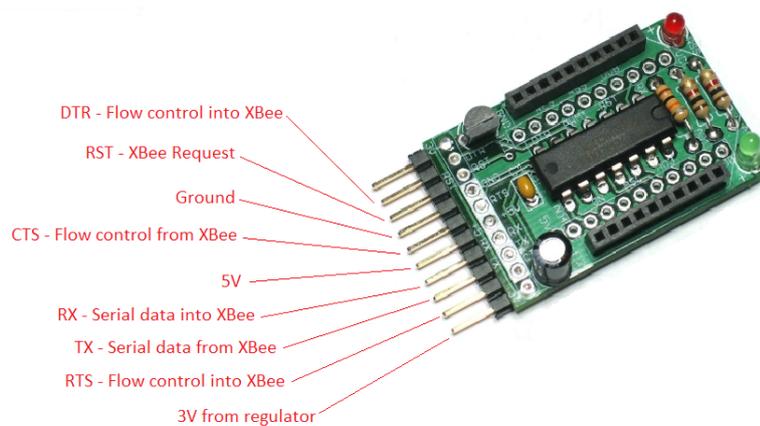


Fig. 6.12: Pinos do adaptador para ligação ao cabo FTDI.

6.3 Configurações dos Módulos XBee

6.3.1 Modos de Operação

Os módulos XBee podem operar de dois modos diferentes: *AT* (*Transparent*) e *API* (*Application Programming Interface*).

Modo AT

Os dados passam através pelo módulo XBee, isto é, os dados recebidos pelo módulo são transmitidos para o XBee de destino. Substitui a linha série, possui uma interface mais simples e as saídas digitais e ADCs não podem ser configurados [59, 72].

Modo API

Neste modo de operação, todos os dados recebidos e enviados são agrupados em pacotes que definem todas as operações do módulo. Isto é, todos os dados essenciais à comunicação são controlados por um *frame*. A Figura 6.13 apresenta a forma como os dados são organizados no *frame*. No seguimento, são descritas as várias partes que o compõem. O modo API possui mais funcionalidades, sendo, contudo, uma interface mais complexa [30].

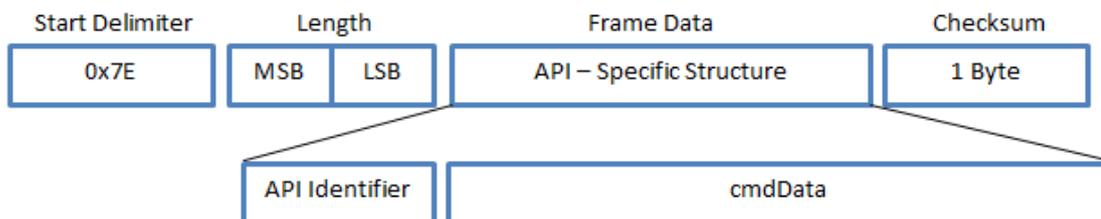


Fig. 6.13: Estrutura geral de um *frame* API.

- *Start Delimiter*: delimitador inicial, corresponde a 1 byte com o valor 0x7E (*Start byte*);
- *Length*: 2 bytes que definem o tamanho do *frame*, MSB (*Most Significant Byte*) e LSB (*Least Significant Byte*). De notar que o último byte, *checksum*, não está incluído no tamanho do *frame*;
- *Frame Data*:
 - *API Identifier*: valor indicativo da função do *frame*, indentificador do *frame*;
 - *cmdData*: conteúdo do *frame* que varia tendo em conta o tipo de comando utilizado;
- *Checksum*: byte responsável por validar a integridade do frame [30].

O formato apresentado anteriormente constitui a forma geral de um pacote API. Todavia, para este trabalho foi utilizado um tipo de pacote mais específico: I/O RX, ou seja, um formato API para uma amostra de dados recebidos RX (*receiver*). Para o objetivo proposto, este pacote é essencial, pois pretende ler-se os dados provenientes de um módulo XBee que funciona como ADC. As especificações do pacote I/O RX

são apresentadas na Figura 6.14 e Tabela 6.4. Nesta fase do trabalho, foi ainda utilizado o guia de informações apresentado no Apêndice C⁶.

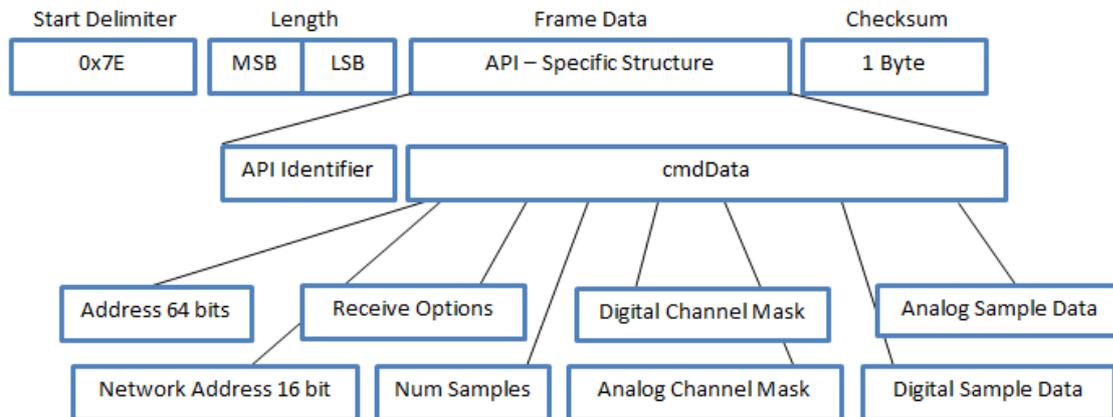


Fig. 6.14: *Frame* para o pacote I/O RX.

É importante nesta fase, fazer uma pequena abordagem à forma como os pinos digitais e analógicos são ativados através das máscaras. Por exemplo, relativamente à *Digital Channel Mask*, esta é constituída por 2 bytes, 8 bits cada, e da seguinte forma:

Primeiro byte: n/a n/a n/a D12 D11 D10 n/a n/a

Segundo byte: D7 D6 D5 D4 D3 D2 D1 D0

Como é possível observar, nem todos os bits correspondem a uma determinada porta digital. Por exemplo, poderá definir-se os dois bytes que compõem a *Digital Channel Mask* como 0x00 e 0x13. Estes bytes, apresentados no formato hexadecimal, são convertidos para o formato binário, resultando em 0000 0000 0000 1101, correspondentes aos 16 bits que constituem a *Digital Channel Mask*. Deste modo, tendo em conta esta sequência de bits, cada número correspondente ao valor 1 dos 2 bytes convertidos para o formato binário, correspondem às portas digitais ativas. Desenvolvido este processo, consegue-se então dizer que as portas digitais ativas para este caso são a D3 D2 e D0.

A máscara *Analog Channel Mask* é definida com recurso a apenas um byte, de 8 bits. O processo é idêntico ao anterior, no entanto, a posição das portas analógicas

⁶ <http://www.tunnelsup.com/>

Tab. 6.4: Especificações para a *frame* I/O RX (fonte: <http://www.tunnelsup.com/>).

Byte	Nome	Descrição
0	Start Byte	Byte que indica o início do <i>frame</i> .
1	Lenght	MSB (<i>Most Significant Byte</i>); LSB (<i>Least Significant Byte</i>)
2		
3	API Identifier	Indica a função da <i>frame</i>
4	Address 64 bits	Byte 4 MSB; Byte 11 LSB
5		
6		
7		
8		
9		
10		
11		
12	Network Address 16 bits	Endereço de rede
13		
14	Receive Options	0x01 <i>Packet Acknowledged</i> ; 0x02 <i>Broadcast packet</i>
15	Num Samples	Número de conjuntos de amostras (sempre 1 devido às limitações do XBee)
16	Digital Channel Mask	Indica que pinos são definidos para DIO (digital I/O)
17		
18	Analog Channel Mask	Indica que pinos são definidos para o ADC
19	Digital Sample Data (se existir)	Se <i>Digital Sample Data</i> >0: cada entrada analógica ativa retorna 2 bytes com o valor medido pelo ADC na entrada analógica
20		
21	Analog Sample data	Se <i>Analog Channel Mask</i> >0: cada entrada analógica ativa retorna 2 bytes com o valor medido pelo ADC na entrada analógica
22		
23	Checksum	Byte responsável por validar a integridade da <i>frame</i>

é diferente, já que existem apenas 4 portas analógicas no módulo XBee:

Byte: n/a n/a n/a A3 A2 A1 A0

Como exemplo, pode definir-se o byte 0x05, que convertido para binário resulta em 0000 0101. Desta forma, as portas analógicas ativas são a A2 e A0.

6.3.2 Configuração no X-CTU

A configuração dos módulos XBee é feita com recurso ao *software* X-CTU ⁷, um programa oficial para configuração dos módulos XBee [69]. Para iniciar a configuração é necessário ligar o módulo XBee a um PC, utilizando um *Shield* ou o adaptador. Depois de ligado o módulo ao PC, o *software* pode ser iniciado, devendo aparecer discriminadas as várias ligações às portas USB do PC. É selecionada a porta a que está ligada o módulo XBee, sendo necessária a definição dos parâmetros associados à mesma. Estes parâmetros são apresentados na Tabela 6.5.

Tab. 6.5: Parâmetros a definir para a porta COM.

Parâmetro	Definição
<i>Baud</i> (Taxa de Transferência)	9600
<i>Flow Control</i> (Controlo de Fluxo)	NONE
<i>Data Bits</i> (Bits de Dados)	8
<i>Parity</i> (Paridade)	NONE
<i>Stop Bits</i> (Bits de paragem)	1

O *software* X-CTU permite realizar um teste de inicialização do dispositivo XBee, de modo a verificar o seu *status*, a versão de *firmware*, bem como se é possível comunicar com o módulo. Após esta fase inicial, é possível configurar os módulos como *Coordinator*, *Router* ou *End-Device*. Para isso deverá seleccionar-se o separador "*Modem Configuration*" e, no menu "*Function Set*", efetuar a seleção pretendida consoante o objetivo específico para cada módulo. Apesar das várias configurações disponíveis, serão abordadas somente as mais relevantes para os testes efetuados.

Para a realização deste trabalho, foram utilizadas duas formas de conexão dos módulos XBee ao PC, para configuração dos mesmos: através do Arduino ou pela utilização do adaptador. No caso do adaptador, este é simplesmente conectado

⁷ <http://www.digi.com/support/>

ao PC através do cabo FTDI. No que se refere ao Arduino, inicialmente é essencial realizar o já referido *bypass* do microcontrolador, retirando-o da placa ou conectando o pin GND ao RESET. Esta ligação exige também a conexão entre os pinos TX e RX no módulo XBee aos correspondentes TX e RX no Arduino. Esta é a única situação em que os pinos TX e RX são ligados aos seus correspondentes. É ainda necessário alimentar o módulo ligando-o aos pinos de alimentação de 5 V e GND, provenientes do Arduino. As ligações efetuadas podem ser visualizadas na Figura 6.15.

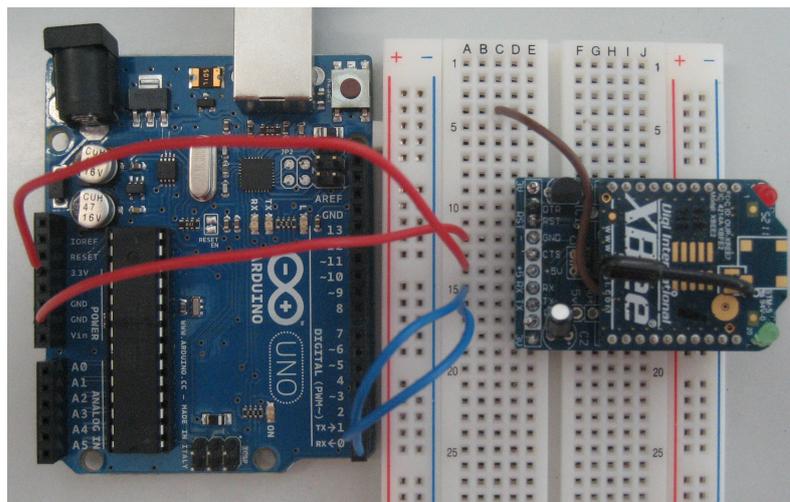


Fig. 6.15: Ilustração das ligações entre o Arduino e módulo XBee, para configuração do módulo com recurso ao *software* X-CTU.

6.4 Testes de Funcionalidade

No sentido de testar a comunicação entre os módulos XBees, foi realizada uma série de testes com vários códigos e diferentes configurações de *hardware*, usando o *software* X-CTU. São apresentados no seguimento apenas alguns dos testes realizados e de maior importância para o presente trabalho.

Ao longo da execução dos testes, é possível identificar fisicamente qual dos dispositivos está definido como ZR ou ZC ou ZED, através da simples observação dos LEDs de *status*. Quando o LED verde pisca uma vez por segundo, o dispositivo foi definido como ZC e, quando pisca 2 vezes por segundo, foi definido como ZR ou ZED. A confirmação da transmissão de dados pode ser verificada através do LED vermelho, quando este se encontra ligado.

Estes testes permitem a concretização do objetivo final deste trabalho, correspondente ao desenvolvimento de um *array* de detetores de radiação ionizante.

6.4.1 Teste 1

A realização deste teste tem por finalidade um *Chat* entre dois módulos XBee. Consiste na transmissão de dados entre um XBee configurado como ZR AT e outro como ZC AT. Desta forma, ambos os módulos estão ligados via USB ao PC para que seja possível através do programa X-CTU enviar mensagens escritas de um para outro, usando uma comunicação via *wireless*. A Figura 6.16 ilustra a configuração das conexões usadas no teste "chat".

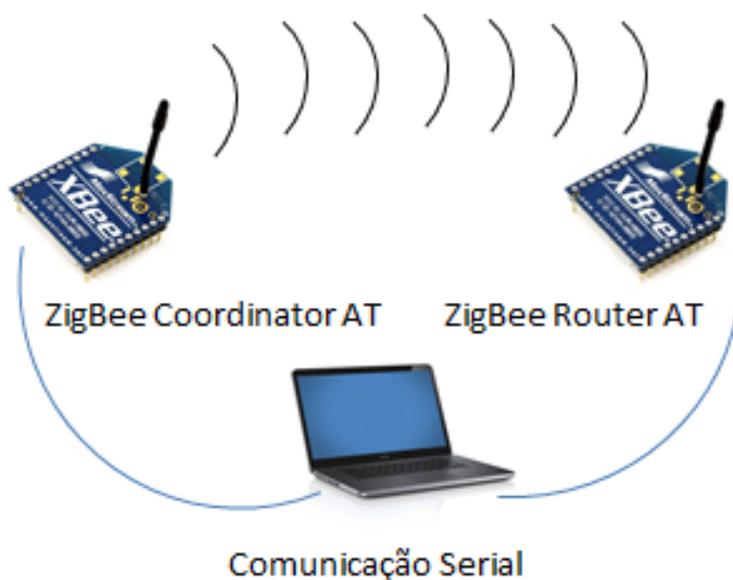
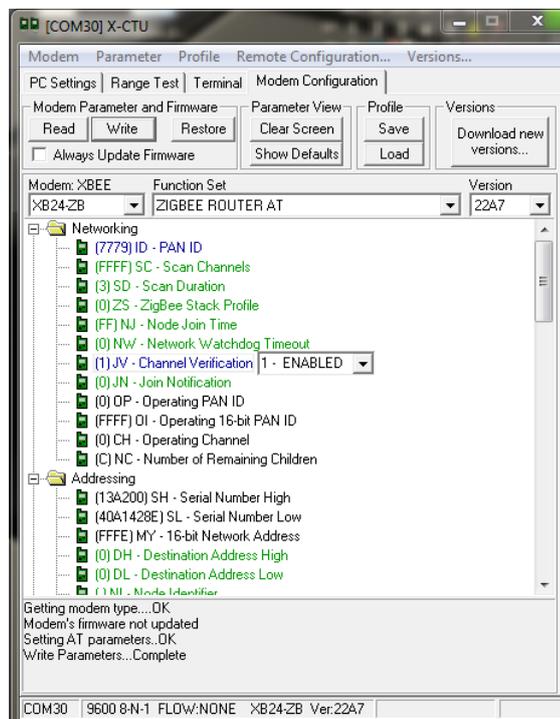


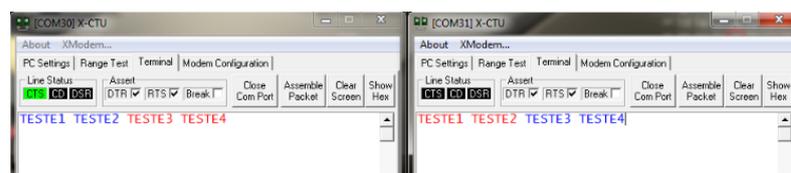
Fig. 6.16: Desenho das conexões para o teste 1 (*Chat*).

Concluídas as conexões, inicia-se o programa *software* X-CTU, confirma-se se os módulos são devidamente reconhecidos pelo programa e seleciona-se o separador *Modem Configuration* para configurar os XBee como ZIGBEE ROUTER AT e ZIGBEE COORDINATOR AT. A Figura 6.17 (a) mostra a aplicação X-CTU com alguns dos parâmetros já definidos para esta configuração. O primeiro passo consiste na leitura dos parâmetros pré-definidos no módulo XBee, usando o botão *Read*; o que permite visualizar todos os dados associados àquele dispositivo. Define-se então que função se pretende atribuir ao dispositivo, no menu *Function Set*, sendo que, neste caso, ficou

como ZR AT. Esta função deverá ser escrita no XBee através do botão *Write*. Posteriormente, é feito um *Restore* para que todos os parâmetros assumam os valores padrão para o ZR AT. É novamente executado um *Read* para confirmação que todos os valores foram realmente restaurados. Primeiramente, deverá definir-se a PAN ID com um qualquer número que terá de ser obrigatoriamente o mesmo para todos os dispositivos da rede a implementar. Para este teste utilizou-se o número 7779. De seguida, convém analisar a verificação de canal (*Channel Verification*) e, caso não esteja definida como ativa (*Enabled*), escolher essa opção. O *Destination Address High* deverá ser definida com o valor padrão, ou seja 0 (*Broadcast*). Efetuadas as alterações necessárias, estas deverão ser gravadas no módulo através da opção *Write*.



(a)



(b)

Fig. 6.17: Configuração e resultados do teste 1 ("Chat"): (a) Configuração X-CTU; (b) Validação da comunicação entre os *modems* XBee.

De seguida, é necessário configurar o outro módulo XBee. Inicia-se novamente o X-CTU e o processo é idêntico ao inicial no que diz respeito à leitura e restauro dos dados no módulo. É agora escolhida a função ZR AT e definida a PAN ID igual à descrita para o ZR (7779). De forma semelhante à realizada para o módulo anterior, deverá ser feito o *Write* para gravar as definições no módulo XBee. Após terminada a configuração de ambos, é possível testar se os módulos estão a comunicar corretamente. Para isso, deve manter-se as duas janelas do X-CTU, correspondentes aos dois dispositivos, abertas simultaneamente no separador *Terminal*. Este terminal possibilita a introdução, pelo utilizador, de comandos e mensagens a partilhar entre os *modems*. As *strings* a azul representam as mensagens enviadas do próprio dispositivo e, a vermelho, as mensagens recebidas pelo outro dispositivo. A Figura 6.17 (b) ilustra a realização deste teste, no qual é possível conferir uma perfeita comunicação entre os dois dispositivos.

6.4.2 Teste 2

Concluídos os testes de comunicação entre os XBee, seguem-se testes mais específicos. Os testes indicados no seguimento foram baseados no uso de dispositivos que geram um sinal analógico ou simplesmente através da ligação a uma tensão de 5 V ou GND, para posterior conversão para um sinal digital. O papel de ADC é neste caso desempenhado pelo o módulo XBee que recebe o sinal analógico. Deste modo, quando o XBee recebe o sinal, envia-o via *wireless* para outro módulo no formato digital. Estes testes foram realizados usando uma topologia de rede *Per*. Após a validação dos testes na topologia *Per* é possível passar para topologias mais complexas, nomeadamente a topologia *Star*.

O teste 2 consiste na configuração dos módulos XBee para a transmissão de sinais analógicos. O primeiro XBee deverá operar no modo API, visto que não será responsável por receber dados diretamente do sensor. Como referido anteriormente, através da função *Write*, este deverá ser configurado como ZC API com uma determinada PAN ID, neste caso 123. Seguidamente, é configurado outro módulo como ZR no modo AT, pois funcionará como conversor ADC, estando ligado diretamente ao detetor de radiação. Para este módulo terão de ser definidos outros parâmetros, além da PAN ID com valor idêntico ao módulo anterior. Inicialmente, deverá colocar-se a opção *Channel Verification* como *Enabled* para garantir que o canal de

comunicação está ativo. De seguida, nas opções de I/O deverá escolher-se uma das portas analógicas apresentadas (AD0, AD1, AD2 ou AD3), por exemplo a AD2, e seleccionar a opção ADC, para que haja conversão dos sinais analógicos em digitais (Figura 6.18). É ainda necessário garantir, *à posteriori* a existência de uma ligação física entre o pino AD2 do adaptador e o dispositivo analógico. Por fim, o parâmetro *Sampling Rate* foi, neste caso, ajustado para o valor 1388, podendo ser obviamente alterado consoante a situação.

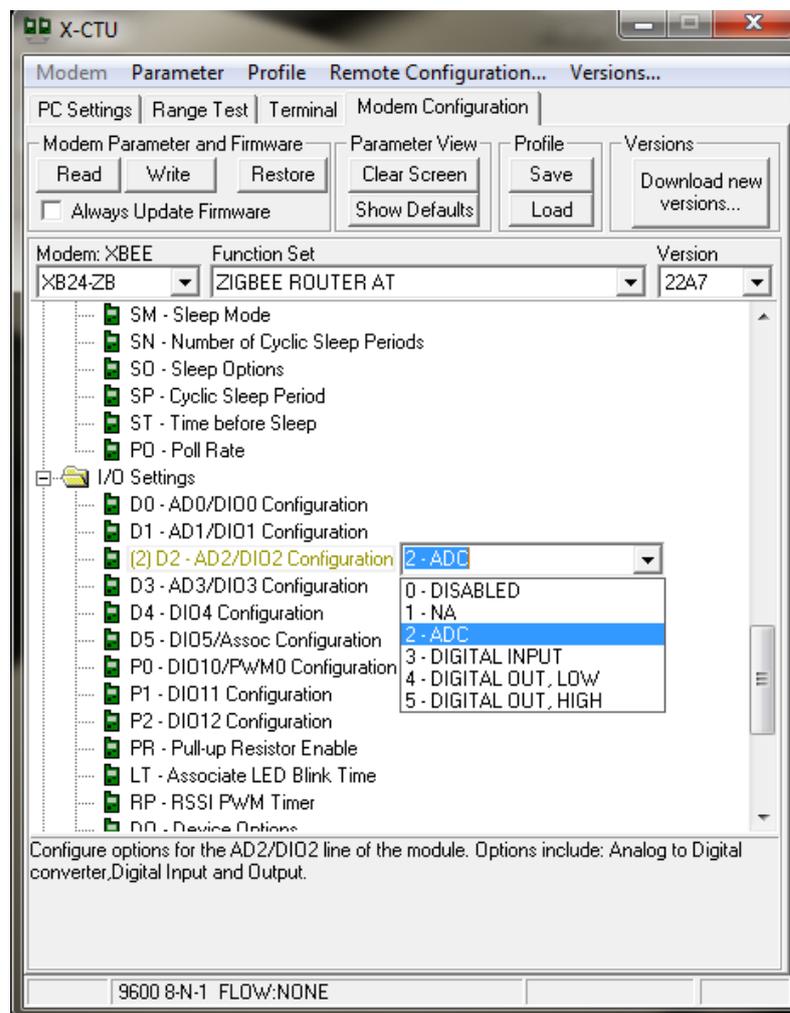


Fig. 6.18: Definições do teste 2: configuração dos parâmetros associados a um módulo XBee como ZR AT, para funcionamento como conversor ADC.

Pela Figura 6.19 (a) é possível perceber a forma como as várias ligações são efetuadas. Contrariamente à situação anterior correspondente à ligação dos módulos ao PC para configuração, os pinos RX e TX do XBee ligam-se aos pinos TX e RX

correspondentes, do Arduino (ver Figura 6.19 (a)). É ainda necessária uma fonte de alimentação externa para alimentar o módulo XBee, bem como o dispositivo analógico ligado a este. Como já foi referido, é nesta fase que é realizada a conexão entre o pino configurado como ADC no X-CTU e o dispositivo analógico, visível a cor castanha na Figura 6.19 (b). Acresce, que a comunicação é confirmada pelo facto do LED vermelho se encontrar permanentemente ligado.

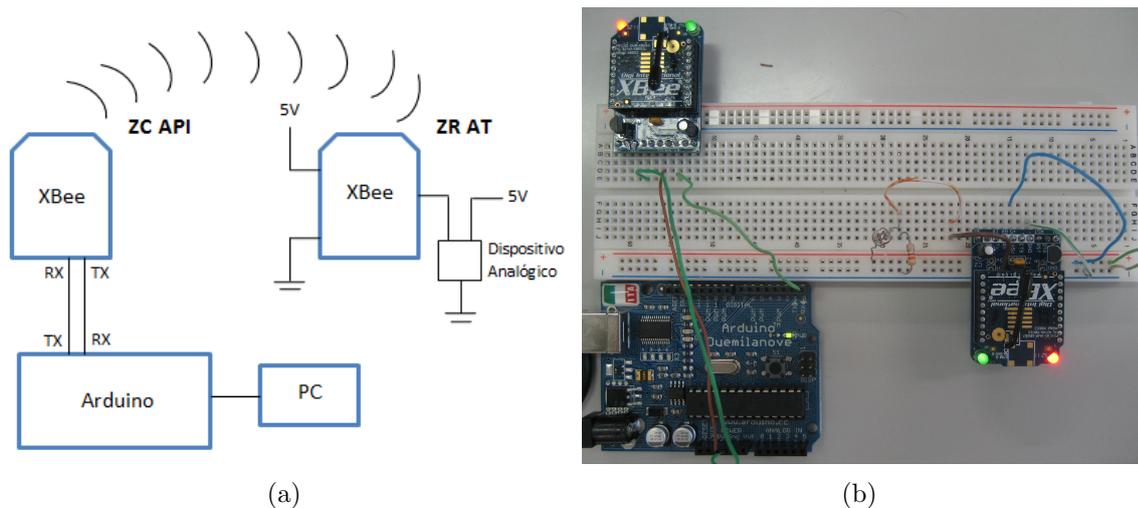


Fig. 6.19: Desenho e implementação do teste 2: (a) Esquema das conexões; (b) Hardware implementado.

6.4.3 Teste 3

Este teste consiste na leitura do pacote de dados no formato hexadecimal proveniente do módulo XBee, alimentado com 0 V e 5 V. Este teste é crucial para visualizar o pacote de dados no formato I/O RX proveniente do ZR, que se liga a uma tensão de 5 V ou ao GND (0 V). Para isso, foi utilizado o código apresentado abaixo. Para dar início à comunicação, é definida a taxa de transmissão de 9600 bits por segundo. De seguida, um comando *if* garante que o ciclo *for* é executado apenas quando o número de bytes na porta série é superior a 21, isto, superior à quantidade de bytes que compõem o pacote. Através do comando *Serial.read()* são lidos os dados que provêm do ZR, no formato hexadecimal (HEX). Cada um destes valores é enviado para o *Serial Monitor* do Arduino, como exemplificado na Figura 6.20.

leiturahexadecimal.ino

```

int readValue=0;

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

void loop(){
  if(Serial.available()>21){
    for(int i=0; i<22; i++){
      Serial.print(Serial.read(),HEX);
      Serial.print(",");
    }
    Serial.println();
  }
}

```

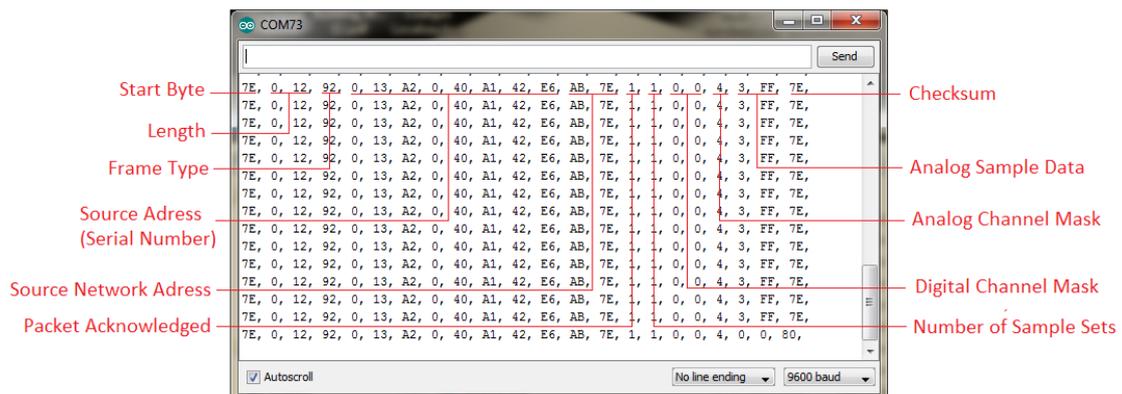


Fig. 6.20: Resultados do teste 3: dados no formato hexadecimal provenientes do módulo XBee conectado a um sinal analógico (0 V e 5 V).

Observando cada linha deste conjunto de dados, podemos identificar os vários bytes que constituem este pacote. O primeiro byte é o *Start byte* que indica o início do *frame* e os dois seguintes dizem respeito ao seu tamanho, que neste caso é 12 em hexadecimal, isto é, 8 bytes no formato decimal. O quarto parâmetro diz respeito ao tipo de *frame*, que neste caso é 92, representativo de uma amostra de dados (*data sample*). Os 8 bytes seguintes correspondem ao *Serial Number* do dispositivo que envia dados para o ZC que os está a receber. Seguem-se, os dois bytes correspondentes à *Source Network Address*. O byte seguinte 1, representando o *Pack*

Acknowledged das *Receive Options*. O número de conjuntos de amostras, *Number of Sample Sets*, é também 1 devido às limitações do XBee, como já foi referido anteriormente. Os dois bytes que se seguem são 0, já que a informação proveniente do ZR é a de que os dados transmitidos provêm de um sinal analógico. Por outro lado, o byte seguinte corresponde ao *Analog Channel Mask* e apresenta o valor 0x04, que no formato binário corresponde a 00000100, como explicado anteriormente, os dados tiveram origem no pino AD2. Tratando-se de dados analógicos, os dois bytes seguintes correspondem ao sinal analógico no formato hexadecimal. Inicialmente, o XBee foi conectado a uma tensão de 5 V, em que os bytes correspondentes são 0x03 e 0xFF e, posteriormente, ao GND onde são registados bytes 0x00 e 0x00, visíveis na última linha do *Serial Monitor* do Arduino (ver Figura 6.20). A vermelho na Figura 6.20, encontram-se devidamente identificados no pacote de dados os vários bytes descritos anteriormente.

6.4.4 Teste 4

Tendo conhecimento da forma como os dados no formato hexadecimal são obtidos, o objetivo seguinte consiste em excluir do pacote de dados todos os bytes desnecessários, ficando apenas com o sinal analógico correspondente à aquisição dos dados dos sensores. Para isso, é necessário a utilização de um código capaz de extrair do *frame* apenas os bytes correspondentes ao *Analog Sample Data*, apresentado abaixo.

Tal como nas situações anteriores, é necessário definir uma taxa de transmissão de 9600 bits. É feita uma verificação inicial para que só se inicie o ciclo *for* quando são detetados na porta série 21 bytes e quando o *Start Byte* (0x7E) é identificado. O ciclo permite que os vários bytes do *frame* sejam descartados até ao byte 18, aproveitando apenas os bytes correspondentes aos dados analógicos. Desta forma, são lidos os dois bytes correspondentes ao sinal analógico (LSB e MSB). O MSB é ainda multiplicado por 256 para que a escala seja ajustada. O resultado deste teste pode ser observado na Figura 6.21, onde é visível a grande alteração de valores, no qual 0 corresponde à ligação do XBee ao GND e aproximadamente 1027 aos 5 V. Este valor máximo deveria corresponder a 1024, no entanto, ocorreu neste caso uma situação de *overhead*, que consiste num processamento ou armazenamento de dados em excesso. É também possível perceber que o valor mais alto não é totalmente

estável, o que poderá estar relacionado com pequenas "falhas na comunicação" ou flutuações na tensão de alimentação.

xbeedados.ino

```
float temp;
void setup(){
  Serial.begin(9600);
};

void loop(){
  if (Serial.available() >= 21){
    if (Serial.read() == 0x7E){
      for (int i=1; i < 19; i++){
        byte discardByte = Serial.read();
      }
      int analogMSB = Serial.read();
      int analogLSB = Serial.read();
      int analogReading = analogLSB + (analogMSB * 256);
      temp = analogReading;
      temp = temp;
      Serial.print(temp);
      Serial.println(" ");
    }
  }
}
```

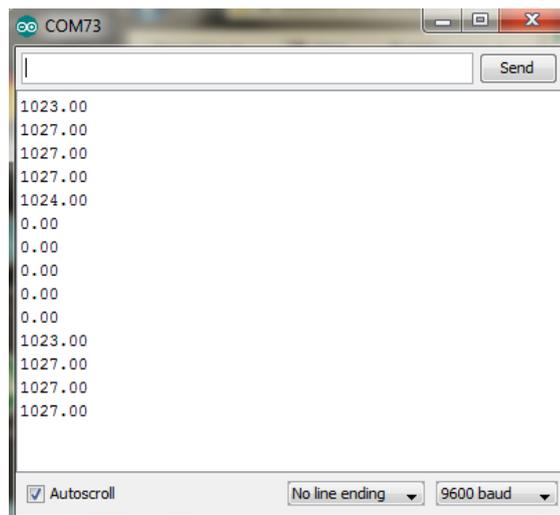


Fig. 6.21: Resultados do teste 4: dados no formato decimal provenientes do módulo XBee conectado ao sinal analógico (0 V e 5 V).

6.4.5 Teste 5

Na realização deste teste, foi usado o mesmo código do caso anterior para o Arduino, as mesmas configurações no X-CTU e os sinais analógicos mantiveram-se em 0 V e 5 V. Foi, no entanto, considerada uma nova configuração do *hardware*. Contrariamente às situações anteriores, não foram utilizados dois adaptadores para os módulos XBee, mas apenas um adaptador e um *Shield* conectado ao PC, como se pode verificar na Figura 6.22. Os resultados obtidos foram idênticos aos apresentados na Figura 6.21, correspondente ao teste 4. Desta forma, concluiu-se que o uso de *Shields* é também viável para o objetivo proposto. Não obstante, os adaptadores são considerados uma melhor opção devido à facilidade de ligação e pelas dimensões reduzidas que apresenta.

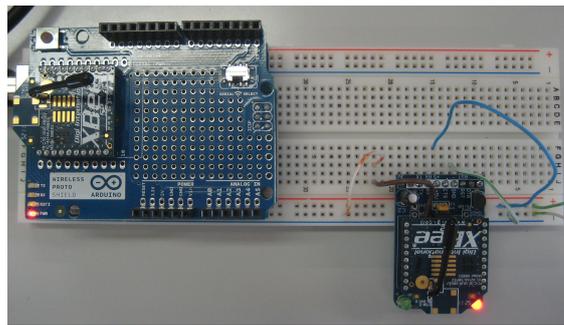


Fig. 6.22: Implementação do teste 4: Comunicação entre XBees com a utilização de um *Shield*, para leitura de sinais analógicos (0 V e 5 V)

6.4.6 Teste 6

Este teste foi efetuado recorrendo ao mesmo *hardware* do caso anterior, com a exceção do uso de uma resistência variável de 100 Ω (Potenciômetro), como pode ser observado na Figura 6.23 (a). Através do uso do potenciômetro pretende gerar-se um sinal analógico variável, semelhante ao sinal que será obtido no circuito eletrónico implementado. A título de exemplo, mostram-se os valores obtidos com o potenciômetro na Figura 6.23 (b), onde é possível observar diferentes valores em função da posição do potenciômetro.

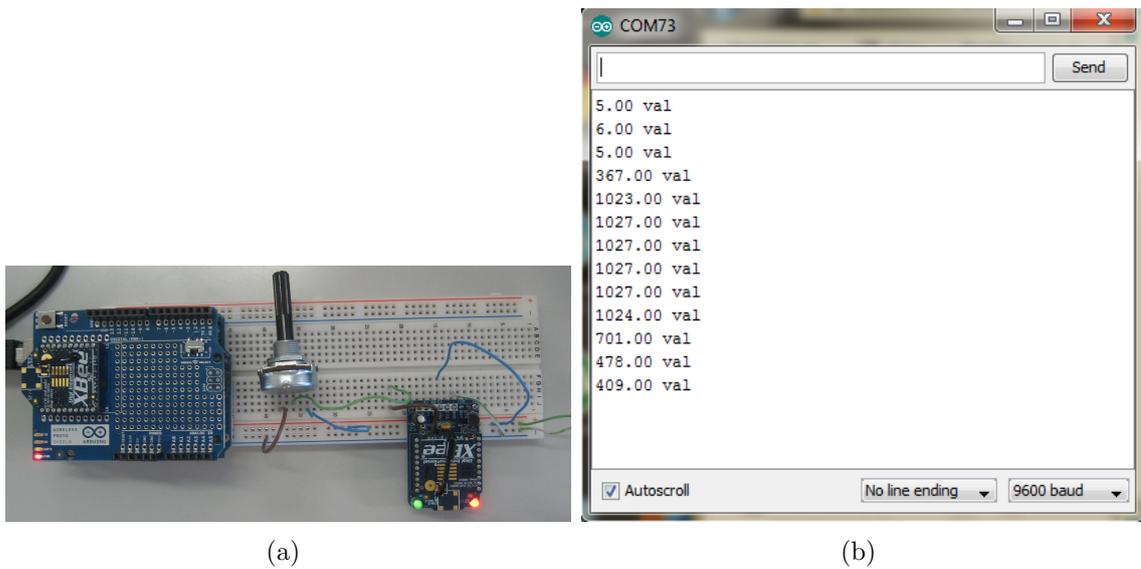


Fig. 6.23: Implementação e resultados do teste 6: (a) Implementação física; (b) Resultados típicos.

6.4.7 Teste 7

Concluídas as fases de testes recorrendo a componentes que permitem gerar sinais analógicos, passou-se aos ensaios finais envolvendo o circuito de deteção ligado ao tubo GM. Nesse sentido, foi necessário fazer a ligação da porta AD2 do adaptador ao circuito, alimentar o módulo e garantir um GND em comum entre ambos. Para alimentar o módulo, seria mais adequado o uso de uma pilha ou um carregador, porém, utilizou-se um outro Arduino, simplesmente para garantir a alimentação do módulo. Na Figura 6.24 é possível verificar a implementação prática deste *hardware*. Com o propósito de adquirir dados para posterior análise, utilizaram-se os mesmos programas anteriormente referidos, para visualização de dados no formato hexadecimal (teste 2) e decimal (teste 3). Na Figura 6.25 são mostrados resultados para estas duas variantes. Na Figura 6.25 (a), a variação dos valores relativos à *Analog Channel Mask* resulta da aproximação de uma fonte radioativa de 3 kBq ao tubo GM. Na Figura 6.25 (b), também é perfeitamente clara a alteração conseguida com base na aproximação da fonte, oscilando entre os 300 e 400.

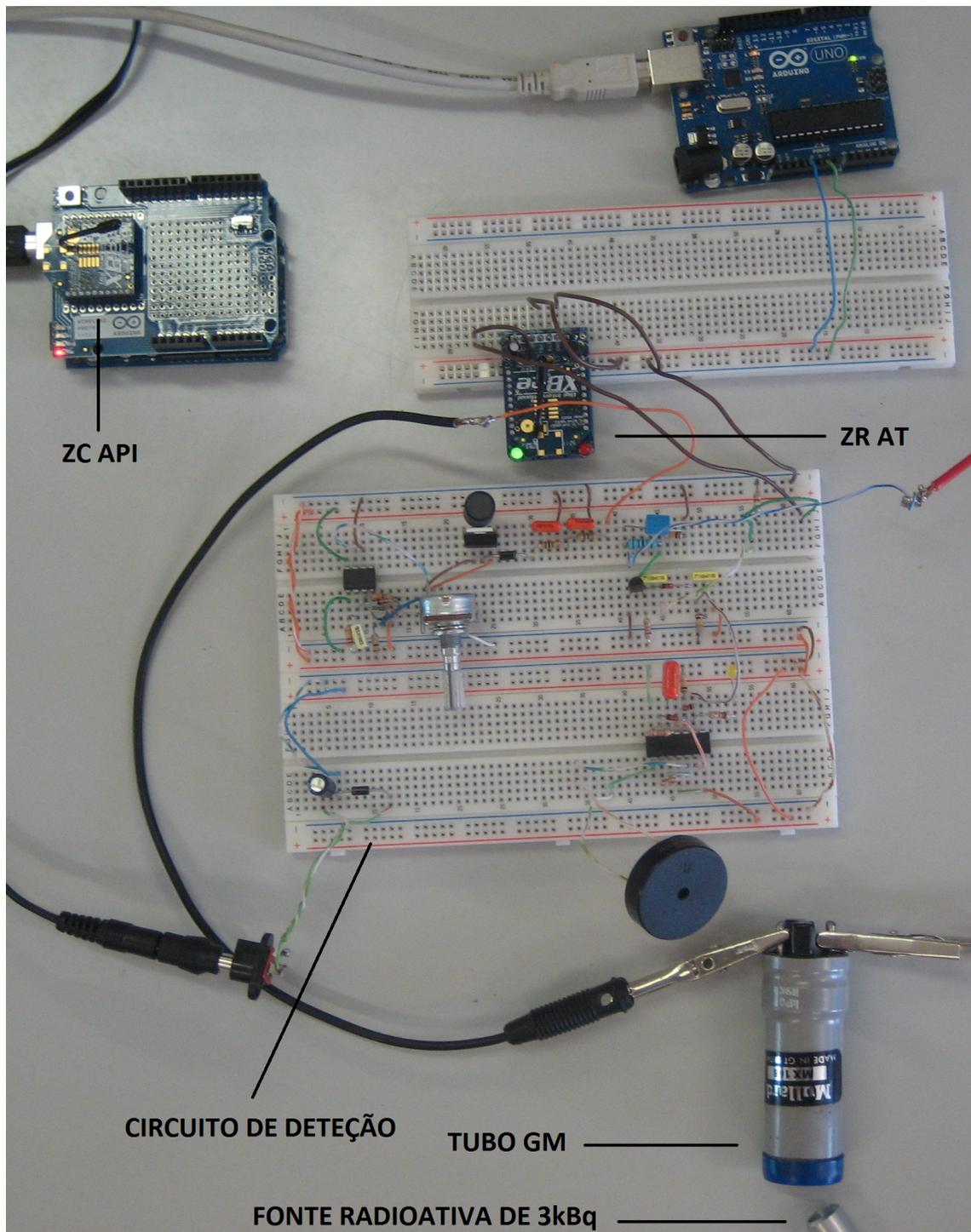
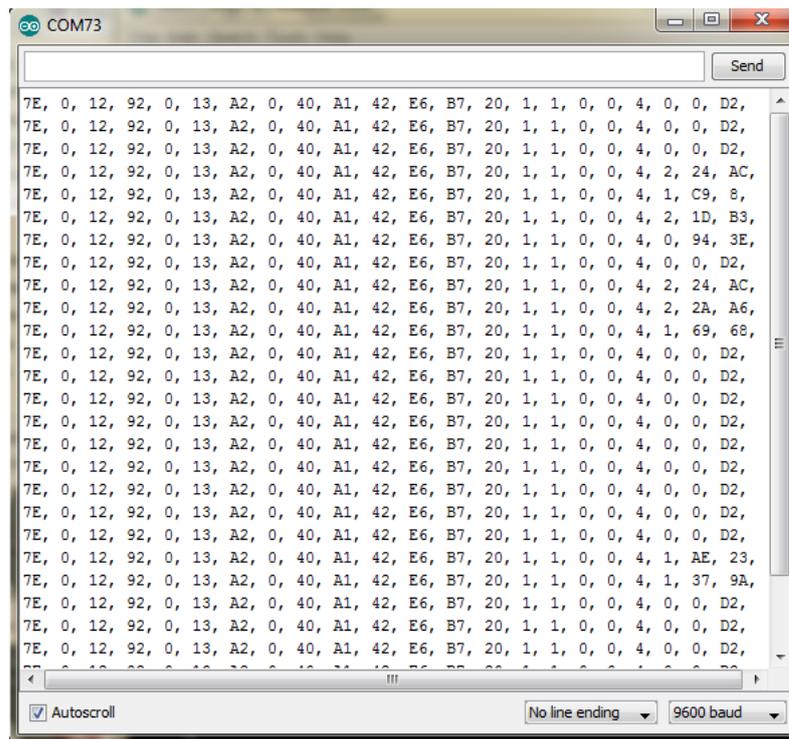
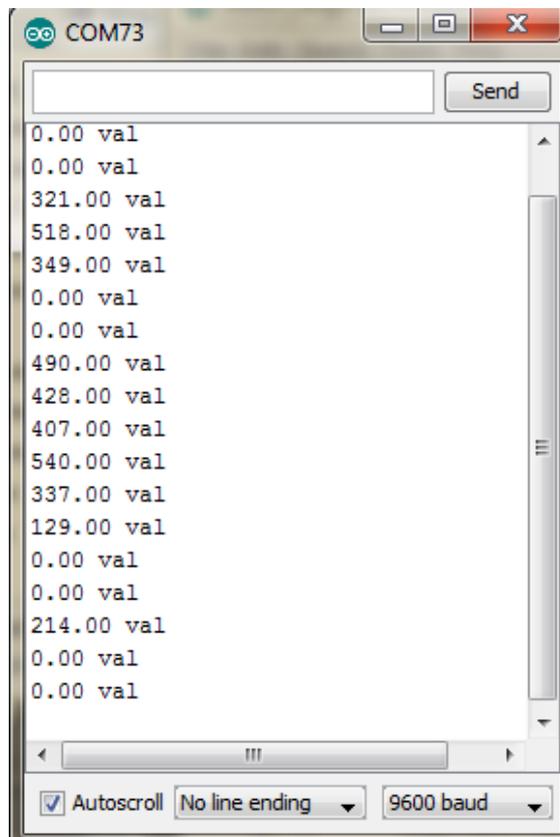


Fig. 6.24: Implementação do teste 7: Hardware para transmissão de dados via *wireless* provenientes do circuito de detecção ligado ao tubo GM.



(a)



(b)

Fig. 6.25: Resultados do teste 7, leituras dos sinais provenientes do circuito de detecção exposto a uma fonte de 3 kBq; (a) Formato hexadecimal; (b) Formato decimal.

6.4.8 Teste 8

Para este teste utilizaram-se três módulos XBee, dois configurados como ZR AT e um como ZC API. Pretende-se que um destes dispositivos se ligue ao GND e o outro a uma tensão de 5V, permitindo assim distinguir os dados enviados pelos dispositivos. Realizadas as ligações observadas na Figura 6.26 (a), foi possível obter os resultados apresentados na Figura 6.26 (b), onde se verifica que os dados são recebidos de forma intercalar, em função do XBee de origem. Cada dispositivo envia um pacote de dados, e o PC processa esses mesmos pacotes de dados, assim sucessivamente. Pretende-se ainda saber qual dos dispositivos está associado a cada um dos valores apresentados, ou seja, usando o *Destination Network Address* é possível discriminar qual deles está a ler um determinado valor. Assim, foi necessária a realização de uma condição que indica qual o sensor que está a ler o valor. A título exemplificativo, resultados deste teste podem ser observados na Figura 6.26 (c).

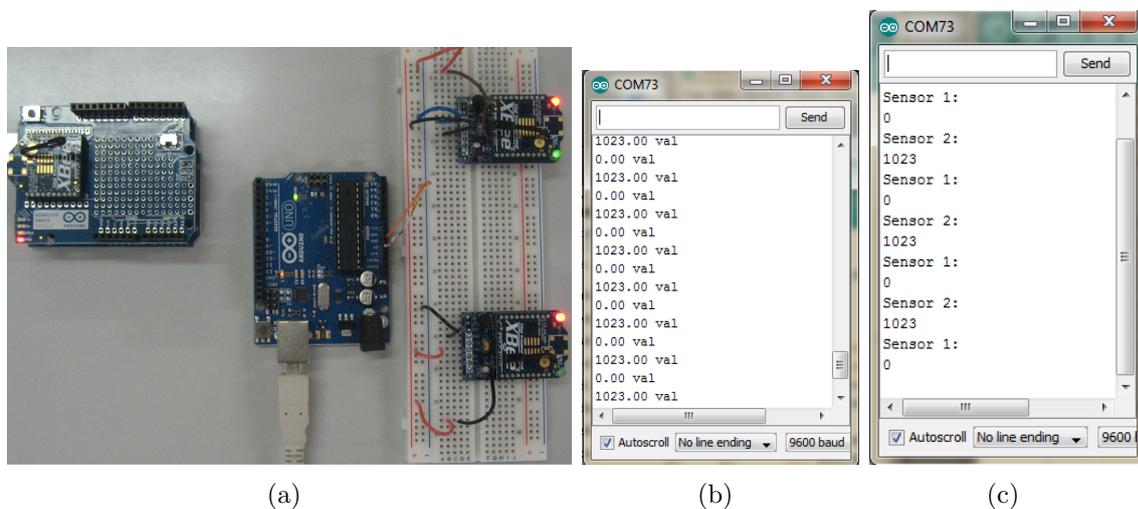


Fig. 6.26: Implementação e resultados do teste 8: (a) Implementação do montagem; (b) Dados em bruto; (c) Discriminação dos sensores.

6.4.9 Teste 9

A iteração final do sistema será constituída por vários sensores capazes de detectar radiação ionizante e transmitir essa informação, via *wireless*, para um PC local.

Desta forma, é possível a monitorização de diferentes espaços, centralizando a informação relativa aos níveis de radiação desses espaços num único local. Na Figura 6.27 é possível observar uma representação esquemática do que se pretende nesta fase. Um PC local deverá ligar-se ao Arduino que se poderá ligar através de um *shield* ou diretamente a um módulo XBee, que, por sua vez, comunica via *wireless* com os restantes dispositivos XBee, conectados aos contadores GM. Este teste foi concebido para validação e observação dos resultados adquiridos no paradigma final. Contudo, limitações de ordem orçamental restringiram o número final de unidades que foram testadas no presente trabalho. A impossibilidade de implementação de um número superior de circuitos esteve relacionada com o facto de os componentes necessários ao desenvolvimento de cada um ser elevado, o que se torna um pouco dispendioso.

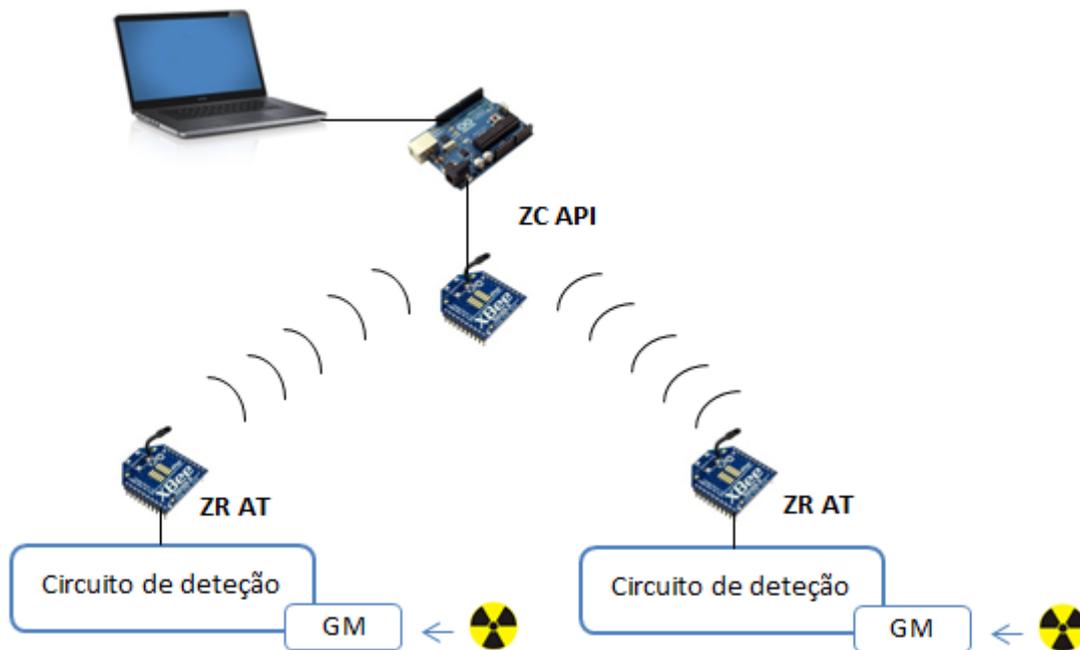


Fig. 6.27: Representação esquemática da implementação final do sistema de detetores de radiação ionizante..

Neste teste, foram utilizados o mesmo número de módulos XBee, bem como uma configuração idêntica. No entanto, cada módulo XBee configurado no modo AT está ligado ao seu próprio circuito de deteção, como foi abordado anteriormente. Na Figura 6.28 é possível observar o sistema implementado, onde é visível a utilização de um *shield* conectado ao PC via USB, dois circuitos de deteção com tubos GM alimentados pela mesma fonte de alimentação de 9 V, e os dispositivos XBee conectados aos circuitos, alimentados pela saída de 5 V de uma placa Arduino Uno. Desta

forma, fez-se a simulação de um pequeno *array* de detetores de radiação ionizante.

Com o intuito diferenciar as contagens provenientes dos dois tubos GM, foi desenvolvido um código semelhante ao apresentado no teste anterior. Além disso, foi ainda desenvolvido um código em Processing com o objetivo de apresentar os dados provenientes de cada sensor num pequeno interface gráfico e posterior gravação em arquivo. Na Figura 6.29 são apresentados alguns resultados, recorrendo a essa mesma interface, considerando três situações distintas no que diz respeito às contagens nos dois tubos GM (Figura 6.29).

Como se pode concluir dos resultados obtidos neste último teste, foi possível validar o funcionamento de um pequeno *array* de sensores de radiação ionizante, capazes de comunicar as suas contagens via *wireless*. Contudo, uma validação mais rigorosa passaria pela adição de um maior número de sensores ao *array*, tendo em vista a implementação de uma rede de monitorização de maiores dimensões. É ainda importante referir que os vários sensores foram colocados em diferentes espaços a diferentes distâncias, para testar a robustez do sistema em termos de comunicações.

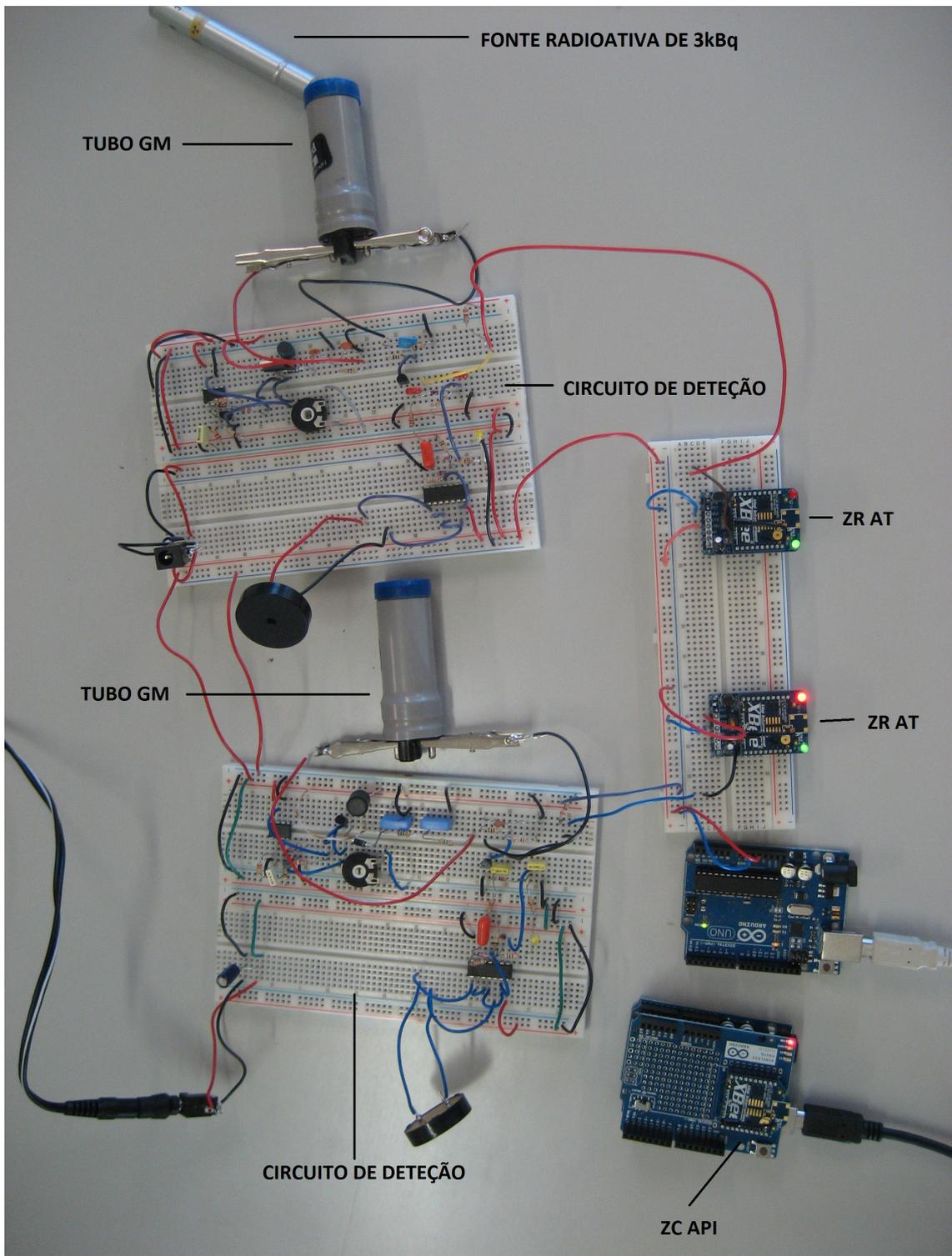
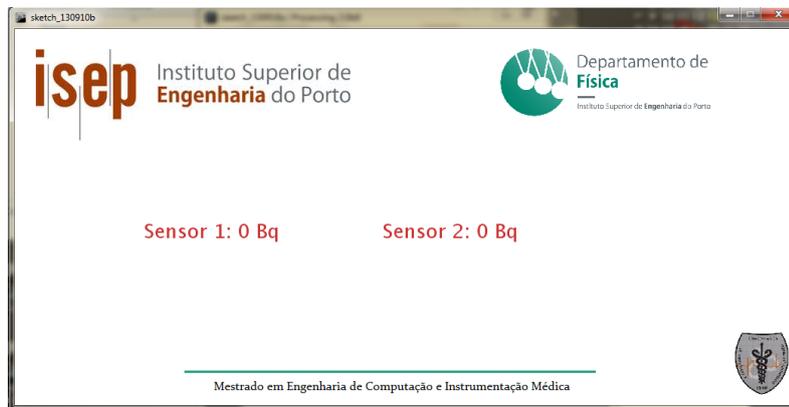
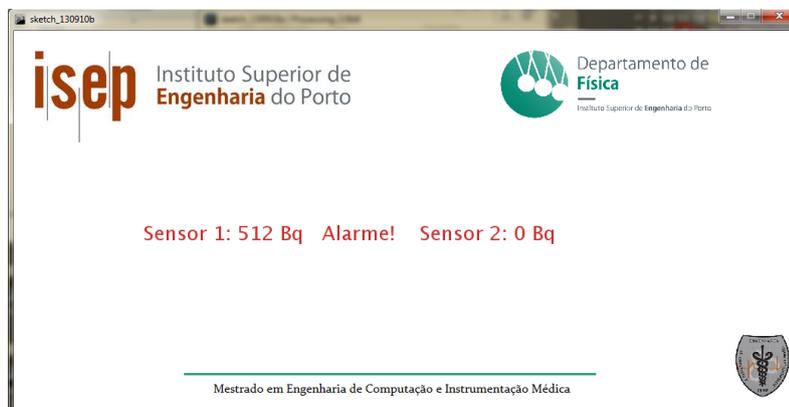


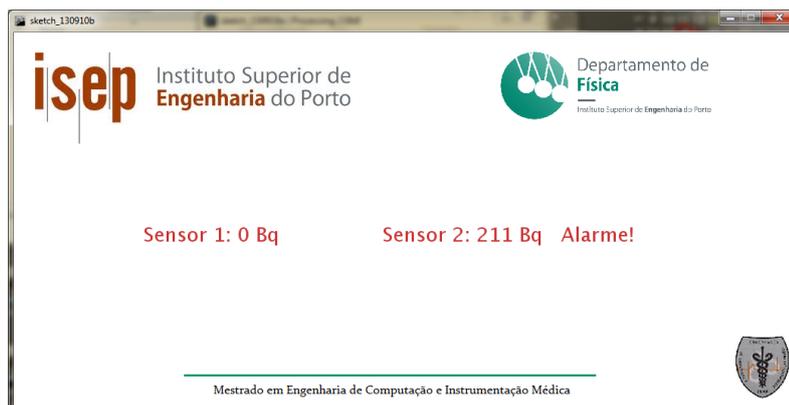
Fig. 6.28: Implementação do sistema de detetores de radiação ionizante.



(a)



(b)



(c)

Fig. 6.29: Interface gráfica simples implementada em Processing, para monitorização das contagens do *array* de detetores: (a) Ausência de radiação ionizante; (b) Fonte radioativa próxima ao Sensor 1; (c) Fonte radioativa próxima ao Sensor 2.

Conclusão

Apesar do elevado potencial de exposição à radiação ionizante na atualidade, nomeadamente em resultado de necessidades de terapia e diagnóstico, existem inúmeras soluções viáveis no que diz respeito à monitorização de espaços físicos com radiação. Neste trabalho, foi apresentada uma solução viável, representada por um sistema constituído por módulos: circuito eletrónico de deteção, aquisição de dados e comunicação sem-fios.

O desenvolvimento do circuito eletrónico obrigou a uma pesquisa acerca dos vários componentes utilizados e a sua funcionalidade no circuito. O circuito foi montado originalmente numa *breadboard*, o que trouxe alguns problemas a nível de robustez do *hardware*. No sentido de desenvolver um protótipo mais robusto, optou-se por soldar o circuito eletrónico numa placa de prototipagem. Apesar de alguns dos resultados com esta montagem serem positivos, o circuito soldado não permitiu a aquisição de dados provenientes do circuito de deteção. É por isso necessário frisar que o processo da soldadura de componentes eletrónicos ser de extremo rigor e cuidado.

A aquisição de dados recorrendo aos conversores analógicos-digitais foi claramente satisfatória. As especificações dos conversores ADC utilizados foram devidamente discrias e as ligações para aquisição de dados apresentadas. Através de testes realizados neste âmbito, foi possível, recorrendo a diferentes ambientes de desenvolvimento integrado, alcançar diferentes formas de visualizar os dados recolhidos, por forma a cumprir os objetivos propostos.

Relativamente às redes sem-fios, optou-se pela utilização da tecnologia ZigBee devido às suas características, nomeadamente o baixo custo, baixo consumo energético e fácil implementação. O estudo pormenorizado de todo o processo de transmissão de dados através de pacotes, constituiu um dos aspetos cruciais para o desenvolvimento do trabalho. Foi ainda possível, criar diferentes configurações de *hardware*, utilizando diferentes sinais analógicos, tipo e número de dispositivos. Foi também possível concluir que a utilização dos adaptadores, constitui uma forma mais simples de configurar os módulos XBee, permitindo substituir os *shields* em algumas situações, nomeadamente em espaços confinados. Os resultados dos testes apresentados comprovam a viabilidade deste tipo de comunicação baseada no protocolo ZigBee, já que foi possível a visualização dos dados provenientes de diferentes módulos XBee, transmitidos para um PC local via *wireless*.

Trabalho futuro

A área das tecnologias sem-fios está em constante evolução e, como tal, muito pode ainda ser melhorado e desenvolvido no decorrer deste trabalho. Testada a comunicação via *wireless* de forma eficaz numa pequena rede com topologia tipo *Star* e *Per*, poderão ser criadas redes contendo um maior número de dispositivos ou baseadas noutras topologias de rede. O ideal seria o desenvolvimento de protótipos para colocar nos vários espaços a monitorizar.

Seria ainda bastante interessante, o desenvolvimento de um interface gráfico com opções relativamente à escolha do sensor que se pretende visualizar, bem como a quantidade exata de radiação que o sensor está a medir em tempo real.

É ainda importante introduzir os sistemas móveis, pois, faria todos o sentido conceber uma pequena aplicação móvel, para que os responsáveis de um determinado espaço sujeito à presença de radiação ionizante, conseguisse, ter acesso aos dados provenientes de cada sensor através de um dispositivo móvel.

Outro aspeto que seria bastante útil, é a criação de um sistema de base de dados onde toda a informação proveniente de cada detetor que constitui o *array* seria armazenada para posterior consulta, consoante as preferências do utilizador.

Bibliografia

- [1] D. Halliday, *Introductory Nuclear Physics*. New York: Wiley, 2nd ed., 1955.
- [2] G. B. Saha, *Physics and Radiobiology of Nuclear Medicine*. Cleveland, USA: Springer, 3rd ed., 2006.
- [3] E. B. Podgorsak, *Biological and Medical Physics, Biomedical Engineering - Radiation Physics for Medical Physicists*. Canada: Springer, 2nd ed., 2013.
- [4] R. Powsner, E. Powsner, *Essencial Nuclear Medicine Physics*. Blackwell Publishing, 2nd ed., 2006.
- [5] J. K. Shultis, R. E. Faw, *Fundamentals of Nuclear Science and Engineering*. Manhattan, Kansas, USA: Marcel Dekker, 1st ed., 2002.
- [6] W. D. Leslie, I. D. Greenberg, *Nuclear Medicine*. Georgetown, Texas, USA: Landes Bioscience, 1st ed., 2003.
- [7] W. E. Burcham, *Nuclear physics: an introduction*. London: Longman, 2nd ed., 1973.
- [8] C. Grupen, B. Shwartz, *Particle Detectors*. New York, USA: Cambridge University Press, 2nd ed., 2008.
- [9] N. Cinkilic, S. K. Cetintas, T. Zorlu, O. Vatan, D. Yilmaz, T. Cavas, S. Tunc, L. Ozkan, R. Bilaloglu, "Radioprotection by two phenolic compounds: Chlorogenic and quinic acid, on X-ray induced DNA damage in human blood lymphocytes in vitro," *Food and Chemical Toxicology*, vol. 53, pp. 359–363, December 2013.
- [10] N. F. Martins, "Uma síntese sobre os efeitos biológicos da radiação ionizante e o papel da alanina utilizada para dosagem de radiações," *Revista de Biologia e Ciências da Terra*, vol. 11, no. 1, 2011.
- [11] N. Tsoulfanidis, *Measurement and Detection of Radiation*. Washington DC, USA: Taylor and Francis, 2nd ed., 1995.

- [12] A. Das, T. Ferbel, *Introduction to Nuclear and Particle Physics*. Singapore: World Scientific, 2nd ed., 2003.
- [13] C. A. F. Santos, “Simulação de Avalanches de Electrões em Detectores Gasosos de Radiação,” Master’s thesis, Universidade de Aveiro - Departamento de Física, Aveiro, 2009.
- [14] S. A. Milinkovic, O. S. Milanko, “Ionization chamber as a gas sensor: organophosphorus compound detection,” *Analytica Chimica Acta*, vol. 331, pp. 233–238, May 1996.
- [15] S. A. Milinkovic, “Application of an ionization chamber as a gas sensor,” *Nuclear Instruments and Methods in Physics Research*, vol. 408, pp. 562–572, November 1998.
- [16] I. Meric, G. A. Johansen, M. B. Holstad, A. F. Calderon, R. P. Garden, “Enhancement of the intrinsic gamma-ray stopping efficiency of Geiger-Muller counters,” *Nuclear Instruments and Methods in Physics Research*, vol. 696, pp. 46–54, August 2012.
- [17] V. G. G. Freitas, “Determinação de dose de radiação, em tempo real, através de inteligência artificial e realidade virtual,” Master’s thesis, Instituto de Engenharia Nuclear, Rio de Janeiro, 2009.
- [18] P. Horowitz, W. Hill, *The Art of Electronics*. Cambridge USA: Cambridge University Press, 2nd ed., 1989.
- [19] D. Lowe, *Electronics for Dummies All-in-one*. New Jersey: John Wiley and Sons, 1st ed., 2012.
- [20] V. C. Meireles, *Circuitos Eléctricos*. Portugal: Lidel, 1st ed., 2007.
- [21] C. Platt, *Make: Electronics*. Canada: O’Reilly, 1st ed., December 2009.
- [22] M. Rahimo, A. Kopta, “The Field Charge Extraction (FCE) Diode A Novel Technology for Soft Recovery High Voltage Diodes,” *International Symposium on Power*, May 2005.
- [23] N. Galster, M. Frecker, E. Carroll, “Application-specific fast-recovery diodes: Design and performance,” tech. rep., Czech Technical University of Prague, Faculty of Electrical Engineering, April 1998.
- [24] Y. Lo, H. Chien, H. Chiu, “Current-input OTRA Schmitt trigger with dual hysteresis modes,” *Internacional Journal of Circuit Theory and Applications*, vol. 38, pp. 739–746, January 2009.
- [25] W. M. Kader, H. Rashid, M. Mamun, M. A. S. Bhuiyan, “Advancement of CMOS Schmitt Trigger Circuits,” *Modern Applied Science*, vol. 6, no. 12, 2012.

-
- [26] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. USA: California Technical Publishing, 2nd ed., 1999.
- [27] M. Banzi, *Getting Started with Arduino*. USA: O'Reilly, 1st ed., October 2009.
- [28] H. Faugel, V. Bobkov, "Open source hard - and software: Using Arduino boards to keep old hardware running," *Fusion Engineering and Design*, 2013.
- [29] D. Wilcher, *Learn Electronics with Arduino*. New York: Apress, 2012.
- [30] M. Hebel, G. Bricker, *Getting Started with XBee RF Modules - A tutorial for BASIC Stamp and Propeller Microcontrollers*. Parallax, 1st ed., 2010.
- [31] R. Dobkin, A. Morgenshtein, A. Kolodny, R. Ginosar, "Parallel vs. Serial On-Chip Communication," tech. rep., VLSI Systems Research Center, Electrical Engineering Department: Technion - Israel Institute of Technology, Haifa, Israel, December 2007.
- [32] T. S. Rappaport, S. Sandhu, "Radio Wave Propagation for Emerging Wireless Personal Communication Systems," tech. rep., Bradley Department of Electrical Engineering - Virginia Polytechnic Institute and State University, Blacksburg, Virginia.
- [33] M. H. Barringer, K. D. Springer, "Radio Wave Propagation," tech. rep.
- [34] R. Prasad, *OFDM for Wireless Communications Systems*. Boston, London: Artech House, 1st ed., 2004.
- [35] A. B. Lugli, D. G. Sobrinho, "Tecnologias Wireless para automação industrial: Wireless Hart, Bluetooth, Wisa, Wi-fi, ZigBee e SP-100," tech. rep., Instituto Nacional de Telecomunicações: Inatel e DG Soluções e Projetos Industriais Ltda., Brasil.
- [36] A. T. Silva, "Módulos de Comunicação Wireless para Sensores," Master's thesis, Universidade do Porto Faculdade de Engenharia, Porto, Julho 2007.
- [37] A. Lisboa, "Rede sem Fio: Estudo Comparativo dos Protocolos Bluetooth e ZigBee aplicados em Ambientes Industriais," tech. rep.
- [38] P. S. S. Chaitanya, S. Agandi, "A Review On Improving Technologies In Wireless Communications," *International Journal of Engineering Trends and Technology*, vol. 4, April 2013.
- [39] F. Monsignore, "Sensoriamento de ambiente utilizando o padrão ZigBee," Master's thesis, Escola de Engenharia de São Carlos - Universidade de São Paulo, São Carlos, 2007.
- [40] J. Lee, Y. Su, C. Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-fi," November 2007.

-
- [41] H. Malafaya, L. Tomás, J. P. Sousa, “Sensorização sem fios sobre ZigBee e IEEE 802.15.4,” tech. rep., Departamento de Engenharia Electrotécnica de Computadores - Faculdade de Engenharia da Universidade do Porto.
- [42] M. Margolis, *Arduino Cookbook*. USA: O’Reilly, 1st ed., March 2011.
- [43] A. G. Smith, *Introduction to Arduino - A piece of cake*. 1st ed., September 2011.
- [44] M. McRoberts, *Arduino Starters Kit Manual - A Complete Begginers Guide to the Arduino*. Earthshine Design, 1st ed., May 2009.
- [45] A. H. Kioumars, L. Tang, “ATmega and XBee-Based Wireless Sensing,” tech. rep., December 2011.
- [46] T. W. Krawczyk Jr., “Circuits for the Design of Serial Communication System Utilizing SiGe HBT Technology,” Master’s thesis, Rensselaer Polytechnic Institute, Troy, New York, November 2000.
- [47] J. Oxeer, H. Blemings, *Practical Arduino - Cool Projects for Open Source Hardware*. Apress, 2009.
- [48] M. L. Hetland, *Beginning Python*. USA: Magnus Lie Hetland, 2nd ed., 2008.
- [49] M. Lutz, *Programming Python*. USA: O’Reilly, 4th ed., 2011.
- [50] M. Dawson, *Python Programming for the absolute beginner*. USA: Course Technology, 3rd ed., 2010.
- [51] J. Noble, *A Designer’s Guide to Processing, Arduino, and openFrameworks - Programming Interactivity*. USA: O’Reilly, 1st ed., July 2009.
- [52] B. W. Evans, *Arduino Programming notebook*. USA: Creative Commons, 1st ed., August 2007.
- [53] D. Beazley, B. K. Jones, *Python Cookbook*. USA: O’Reilly, 3rd ed., 2013.
- [54] Q. Pan, J. Wu, Y. Wang, J. Ni, “A Effective Method for Properties Access in MAC Layer of ZigBee,” *Communication and Network*, vol. 3, pp. 149–152, June 2011.
- [55] Z. Sahinoglu, P. Orlik, J. Zhang, B. Bhargava, G. Ding, “Reliable Broadcasting in ZigBee Networks,” tech. rep., Mitsubishi Electric Research Laboratories, Cambridge, Massachusetts, December 2005.
- [56] V. Mayalarp, N. Limpaswadpaisarn, T. Poombansao, S. Kittipiyakul, “Wireless Mesh Networking with XBee,” tech. rep., School of Information, Computer, and Communication Technology - Sirindhorn International Institute of Technology, Pathumthani, Thailand.

- [57] K. Masica, *Recommended Practices Guide For Securing ZigBee Wireless Networks in Process Control System Environments*. Homeland Security, April 2007.
- [58] S. C. Ergen, “ZigBee/IEEE 802.15.4 Summary,” tech. rep., September 2004.
- [59] A. S. Rivero, “Rede de Sensores sem Fio para Monitoramento de Equipamentos Eletrônicos,” Master’s thesis, Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, 2011.
- [60] V. P. Rao, D. Marandin, “Adaptive Backoff Exponent Algorithm for Zigbee (IEEE 802.15.4),” tech. rep., Technische Universität Dresden, Germany.
- [61] A. V. Nieuwenhuyse, M. Alves, A. Koubâa, “On the use of the ZigBee protocol for Wireless Sensor Networks,” tech. rep., Polytechnic Institute of Porto, School of Engineering, Porto, Portugal, June 2006.
- [62] E. Feo, G. A. Di Caro, “An analytical model for IEEE 802.15.4 non-beacon enabled CSMA/CA in multihop wireless sensor networks,” tech. rep., Dalle Molle Institute for Artificial Intelligence, Lugano, Switzerland.
- [63] “ZigBee for Applications Developers,” tech. rep.
- [64] S. Naseer, S.R. Chaudhry, “LR-WPAN Formation Topologies using IEEE 802.15.4,” *International Journal of Computer Science Issues*, vol. 8, pp. 1694–1814, November 2011.
- [65] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, Y. F. Hu, “Wireless sensor networks A survey on the state of the art and the 802.15.4 and ZigBee standards,” *Computer Communications*, vol. 30, pp. 1655–1595, December 2007.
- [66] L. Jaiswal, J. Kaur, G. Singh, “Performance Analysis of Backoff Exponent Behaviour at MAC Layer In ZigBee Sensor Networks,” *International Journal of Computer Applications*, vol. 57, pp. 975–887, November 2012.
- [67] R. Morais, M. A. Fernandes, S. G. Matos, C. Serôdio, P. J. S. G. Ferreira, M. J. C. S. Reis, “A ZigBee multi-powered wireless acquisition device for remote sensing applications in precision viticulture,” *Computers and Electronics in Agriculture*, no. 62, pp. 94–106, 2008.
- [68] A. Koubâa, A. Cunha, M. Alves, E. Tovar, “TDBS: a time division beacon scheduling mechanism for ZigBee cluster-tree wireless sensor networks,” tech. rep., Polytechnic Institute of Porto, School of Engineering, Porto, Portugal, 2008.
- [69] R. Faludi, *Building Wireless Sensor Networks*. USA: O’Reilly, 1st ed., December 2010.

- [70] S. G. S. P. Yadav, A. Chitra, “Wireless Sensor Networks - Architectures, Protocols, Simulators and Applications: a Survey ,” *International Journal of Electronics and Computer Science Engineering*, vol. 1, no. 4, pp. 2277–1956.
- [71] Digi International Inc., *XBee/XBee-PRO ZB RF Modules*, November 2010.
- [72] R. Faludi, J. Husney, “ICTP: Wireless Sensor Networks Workshop,”

Apêndice **A**

Aquisição: Código Teste 2

Este anexo corresponde ao código realizado em Processing para leitura de dados da porta série e criação de um gráfico com os dados recolhidos, em tempo real.

detetor.pde

```
import processing.serial.*;

Serial myPort;
String buff = "";
int NEWLINE = 10;

int graphXPos = 1;
int graphXPos_last = 0;
int val_last = 0;
int val;
int distance;

PrintWriter output;

void setup ()
{
  size(1000, 512);

  String portName = "COM29";
  myPort = new Serial(this, portName, 9600);

  background(0, 0, 0);

  output = createWriter("temp.txt");
}

void draw () {
```

```

    fill(255);
    text("Tensao:",          20, height - 10);
    fill(100, 255, 255);
    text(distance,          100, height - 10);

    while (myPort.available() > 0) { serialEvent(myPort.read()); }
}

void serialEvent(int serial)
{

    if(serial != NEWLINE) {
        buff += char(serial);
    } else {

        String sRange = "";
        sRange = buff.substring(0, buff.length() - 1);
        distance = Integer.parseInt(sRange) / 2;
        buff = "";
    }

    int val = height - distance;

    stroke(0);
    fill(0);
    rect(graphXPos, 0, 10, height);
    noFill();
    stroke(128);
    point(graphXPos, height / 2);

    stroke(255, 255, 255);
    line(graphXPos, val, graphXPos_last, val_last);

    output.println(distance);
    output.flush();

    println(distance);

    if (graphXPos > width) {
        graphXPos = 1;
        graphXPos_last = 0;
    }
    else {
        val_last = val;
        graphXPos_last = graphXPos;
        graphXPos++;
    }
}

```

```
    }  
}  
  
void keyPressed() {  
  output.close();  
  exit();  
}
```


Apêndice **B**

Código Ascii

Neste Anexo é apresentado o código Ascii (*American Standard Code for Information Interchange*). Cada símbolo do código Ascii corresponde a um determinado caractere utilizado na escrita (8 bits ou 1 byte).

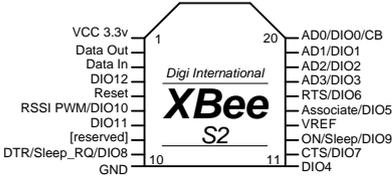
Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	€#32;	Space	64	40	100	€#64;	⓪	96	60	140	€#96;	`
1	1	001	SOH (start of heading)	33	21	041	€#33;	!	65	41	101	€#65;	A	97	61	141	€#97;	a
2	2	002	STX (start of text)	34	22	042	€#34;	"	66	42	102	€#66;	B	98	62	142	€#98;	b
3	3	003	ETX (end of text)	35	23	043	€#35;	#	67	43	103	€#67;	C	99	63	143	€#99;	c
4	4	004	EOT (end of transmission)	36	24	044	€#36;	\$	68	44	104	€#68;	D	100	64	144	€#100;	d
5	5	005	ENQ (enquiry)	37	25	045	€#37;	%	69	45	105	€#69;	E	101	65	145	€#101;	e
6	6	006	ACK (acknowledge)	38	26	046	€#38;	&	70	46	106	€#70;	F	102	66	146	€#102;	f
7	7	007	BEL (bell)	39	27	047	€#39;	'	71	47	107	€#71;	G	103	67	147	€#103;	g
8	8	010	BS (backspace)	40	28	050	€#40;	(72	48	110	€#72;	H	104	68	150	€#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	€#41;)	73	49	111	€#73;	I	105	69	151	€#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	€#42;	*	74	4A	112	€#74;	J	106	6A	152	€#106;	j
11	B	013	VT (vertical tab)	43	2B	053	€#43;	+	75	4B	113	€#75;	K	107	6B	153	€#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	€#44;	,	76	4C	114	€#76;	L	108	6C	154	€#108;	l
13	D	015	CR (carriage return)	45	2D	055	€#45;	-	77	4D	115	€#77;	M	109	6D	155	€#109;	m
14	E	016	SO (shift out)	46	2E	056	€#46;	.	78	4E	116	€#78;	N	110	6E	156	€#110;	n
15	F	017	SI (shift in)	47	2F	057	€#47;	/	79	4F	117	€#79;	O	111	6F	157	€#111;	o
16	10	020	DLE (data link escape)	48	30	060	€#48;	0	80	50	120	€#80;	P	112	70	160	€#112;	p
17	11	021	DC1 (device control 1)	49	31	061	€#49;	1	81	51	121	€#81;	Q	113	71	161	€#113;	q
18	12	022	DC2 (device control 2)	50	32	062	€#50;	2	82	52	122	€#82;	R	114	72	162	€#114;	r
19	13	023	DC3 (device control 3)	51	33	063	€#51;	3	83	53	123	€#83;	S	115	73	163	€#115;	s
20	14	024	DC4 (device control 4)	52	34	064	€#52;	4	84	54	124	€#84;	T	116	74	164	€#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	€#53;	5	85	55	125	€#85;	U	117	75	165	€#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	€#54;	6	86	56	126	€#86;	V	118	76	166	€#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	€#55;	7	87	57	127	€#87;	W	119	77	167	€#119;	w
24	18	030	CAN (cancel)	56	38	070	€#56;	8	88	58	130	€#88;	X	120	78	170	€#120;	x
25	19	031	EM (end of medium)	57	39	071	€#57;	9	89	59	131	€#89;	Y	121	79	171	€#121;	y
26	1A	032	SUB (substitute)	58	3A	072	€#58;	:	90	5A	132	€#90;	Z	122	7A	172	€#122;	z
27	1B	033	ESC (escape)	59	3B	073	€#59;	;	91	5B	133	€#91;	[123	7B	173	€#123;	{
28	1C	034	FS (file separator)	60	3C	074	€#60;	<	92	5C	134	€#92;	\	124	7C	174	€#124;	
29	1D	035	GS (group separator)	61	3D	075	€#61;	=	93	5D	135	€#93;]	125	7D	175	€#125;	}
30	1E	036	RS (record separator)	62	3E	076	€#62;	>	94	5E	136	€#94;	^	126	7E	176	€#126;	~
31	1F	037	US (unit separator)	63	3F	077	€#63;	?	95	5F	137	€#95;	_	127	7F	177	€#127;	DEL

Fig. B.1: Código Ascii (fonte: <http://www.asciitable.com/>).

Apêndice **C**

Guia de Informações XBee

O presente anexo corresponde a um guia de informações acerca da configuração e formato de envio de dados dos módulos XBee. As informações aqui reunidas foram essenciais no decorrer do trabalho.

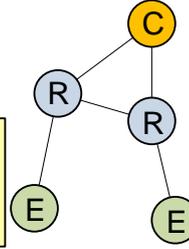


XBee S2 Quick Reference Guide

IEEE 802.14.5 = Zigbee Protocol. XBee is a microcontroller made by digi which uses the Zigbee protocol. The XBee uses 3.3V and has a smaller pin spacing than most breadboards/proto boards. Because of this, it is often useful to purchase a kit to interface the XBee with a breadboard.

Sept/2012 <http://tunnelsup.com>

Specs	Operating Voltage: 2.1 – 3.6V Operating Current: 40mA@3.3V Indoor range: 40 Meters Line of sight range: 120 Meters Max Analog Pin Reading: 1.2V	Digital I/O pins: 11 Analog input pins: 4 Mesh routable Self Healing network Firmware: ZB ZigBee	RF Data Rate: 250kbps Throughput speed: 35kbps Frequency: ISM 2.4GHz OK Temp: -40 to 85C
-------	---	--	---



XBee Roles	Coordinator – 1 required in every network In charge of setting up the network Can never sleep
Router – multiple may exist Can relay signals from other routers/EPs Can never sleep	
End Point – multiple may exist Cannot relay signals Can sleep to save power	

XBee Modes	Transparent – Communication through the XBee. If data is not generated from the XBee itself then both XBee's should be set to AT. Command – Communication to the XBee. If one XBee is sensing data, that XBee should be in AT mode while the receiving one should be in API mode.
------------	--

Arduino Connectivity:	Arduino TX connects to XBee RX (Data in) Arduino RX connects to XBee TX (Data out)
-----------------------	---

XBee Setup	Connect the XBee to a TTL Serial FTDI adapter – OR – Arduino hack: Connect RX to RX, TX to TX, RESET to ground to bypass the Arduino entirely and get serial to XBee. Use the free X-CTU software to configure the XBee. Baud: 9600 – FC: Hardware – Data Bits: 8 – Parity: None – Stop Bits: 1
------------	---

Arduino Integration:	Data sent to Serial.print() will go out TX port of Arduino which is then connected to the RX port of XBee. If XBee is in AT mode it will transmit it wirelessly. Data received from XBee will be sent to the Serial.
----------------------	--

Basic Settings	PAN ID – The network to communicate over. If 0, the XBee will join any. DH/DL – Destination Serial number. Used to send to a specific XBee's Serial. Set to 0 to send to just the Coordinator. Set to 0x000000000000FFFF to broadcast. JV – Router/EP should be set to 1 so it rejoins the network on startup
----------------	---

Arduino Example: Read an analog value using API	<pre>// Remote XBee: AT, Base XBee: API if (Serial.available() >= 21) { // Make sure the frame is all there if (Serial.read() == 0x7E) { // 7E is the start byte for (int i = 1; i < 19; i++) { // Skip ahead to the analog data byte discardByte = Serial.read(); } int analogMSB = Serial.read(); // Read the first analog byte data int analogLSB = Serial.read(); // Read the second byte int analogReading = analogLSB + (analogMSB * 256); } }</pre>
---	--

Pin Settings	For pin settings to work, receiver XBee must be in API mode D0 – Set pin 0 to start sensing IR – Collect data on sensing pins every XX millisecs
--------------	---

Arduino Example: Change the pin setting on a remote Xbee	<pre>// Remote XBee: AT, Base XBee: API Serial.write(0x7E); // Sync up the start byte Serial.write((byte)0x0); // Length MSB (always 0) Serial.write(0x10); // Length LSB Serial.write(0x17); // 0x17 is the frame ID for sending an AT command Serial.write((byte)0x0); // Frame ID (no reply needed) Serial.write((byte)00); // Send the 64 bit destination address Serial.write((byte)00); // (Sending 0x000000000000FFFF (broadcast)) Serial.write((byte)00); Serial.write((byte)00); Serial.write((byte)00); Serial.write(0xFF); Serial.write(0xFF); Serial.write(0xFF); // Destination Network Serial.write(0xFE); // (Set to 0xFFFFE if unknown) Serial.write(0x02); // Set to 0x02 to apply these changes Serial.write('D'); // AT Command: D1 Serial.write('1'); Serial.write(0x05); // Set D1 to be 5 (Digital Out HIGH) long chexsum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'D' + '1' + 0x05; Serial.write(0xFF - (chexsum & 0xFF)); // Checksum</pre>
--	--

API format for Remote AT Command Request	Byte	Example	Description
0	0x7e		Start byte – Indicates beginning of data frame
1	0x00		Length – Number of bytes (ChecksumByte# – 1 – 2)
2	0x10		
3	0x17		Frame type - 0x17 means this is a AT command Request
4	0x52		Frame ID – Command sequence number
5	0x00		64-bit Destination Address (Serial Number)
6	0x13		MSB is byte 5, LSB is byte 12
7	0xA2		
8	0x00		0x0000000000000000 = Coordinator
9	0x40		0x0000000000000000 = Broadcast
10	0x77		
11	0x9C		
12	0x49		
13	0xFF		Destination Network Address
14	0xFE		(Set to 0xFFFFE to send a broadcast)
15	0x02		Remote command options (set to 0x02 to apply changes)
16	0x44 (D)		AT Command Name (Two ASCII characters)
17	0x02 (2)		
18	0x04		Command Parameter (queries if not present)
19	0xF5		Checksum

API format for I/O Data Sample RX Indication	Byte	Example	Description
0	0x7e		Start byte – Indicates beginning of data frame
1	0x00		Length – Number of bytes (ChecksumByte# – 1 – 2)
2	0x14		
3	0x92		Frame type - 0x92 indicates this will be a data sample
4	0x00		64-bit Source Address (Serial Number)
5	0x13		MSB is byte 4, LSB is byte 11
6	0xA2		
7	0x00		
8	0x40		
9	0x77		
10	0x9C		
11	0x49		
12	0x36		Source Network Address – 16 Bit
13	0x6A		
14	0x01		Receive Opts. 01=Packet Acknowledged. 02=Broadcast packet
15	0x01		Number of sample sets. Always set to 1 due to XBEE limitations
16	0x00		Digital Channel Mask – Indicates which pins are set to DIO
17	0x20		
18	0x01		Analog Channel Mask – Indicates which pins are set to ADC
19	0x00		Digital Sample Data (if any) – Reads the same as Digital Mask
20	0x14		
21	0x04		Analog Sample data (if any)
22	0x25		There will be two bytes here for every pin set for ADC
23	0xF5		Checksum(0xFF - the 8 bit sum of the bytes from byte 3 to this byte)

Sleep Mode	Endpoints can sleep to save power. A endpoint that only wakes up every 5 minutes to send data may only be awake for 6 seconds a day. SM – 4 = Cyclic sleep SP – Sleep time (up to 28 secs) SN – Number of sleep cycles ST – Time awake
------------	--

Pin I/O Options	0 – Disabled 1 – N/A 2 – ADC 3 – Digital IN 4 – Digital OUT, LOW 5 – Digital OUT, HIGH
-----------------	---

Digital Ch Mask	First Byte n/a n/a n/a D12 D11 D10 n/a n/a Second Byte D7 D6 D5 D4 D3 D2 D1 D0 Example: 0x00 0x13 = 0000 0000 0000 1101 Pins D3, D2 and D0
Analog Ch Mask	(volt) n/a n/a n/a A3 A2 A1 A0 Example: 0x05 = 0000 0101 = Pin A2 and A0