

António Júlio de Almeida Proença Barata

Protótipo Laboratorial de um Microscópio de Varrimento Usando um Sensor Linear: Avaliação da Sua Resposta Axial



Dissertação para obtenção de grau de Mestre em **Instrumentação e Microelectrónica**,
especialização em **Projecto de Sistemas de Instrumentação**, pela **Faculdade de
Ciências e Tecnologia da Universidade de Coimbra**

Coimbra, 21 de Janeiro de 2008

Resumo

O objectivo do presente trabalho consistiu em criar um sistema laboratorial que envolva diferentes áreas como a electrónica, óptica, processamento de sinal e software para tornar possível o uso de um sensor linear como detector num microscópio óptico de varrimento (*SOM - Scanning Optical Microscope*) avaliando a sua confocalidade.

Foi construído um protótipo de baixo custo de um sistema *SOM* com configuração em reflexão epi-iluminada e a utilização do varrimento da plataforma do objecto para minimizar as aberrações causadas pelo uso de componentes ópticos de baixo custo. Foram usados três circuitos ópticos diferentes e escolheu-se o melhor avaliando a resolução axial e lateral sem o uso de pinholes. A iluminação em linha foi obtida pelo uso de uma lente óptica anamórfica de baixo custo. O sistema óptico, o sistema de aquisição (com base no microcontrolador *PIC* e no sensor *CMOS LIS1024* integrados num circuito impresso), o sistema de visualização e respectivo software estão também aqui descritos.

Os resultados inicialmente obtidos foram adquiridos usando como objecto um espelho plano montado numa plataforma de transacção. Foram utilizadas rotinas em *Matlab* para avaliar a resolução axial e a confocalidade do sistema.

Abstract

The propose of this work is to establish a laboratory system that embraces areas such as electronics, optics, signal processing and software to make possible the application of a linear sensor as detector in a scanning optical microscope (*SOM*) and to evaluate those confocality.

Low-cost *SOM* laboratory prototype with reflection epiilluminated configuration was assembled and a stage scanning type was selected to minimize the aberrations because low-cost optical components were employed. Three different optical circuits were used to test and to choose the best of them evaluating the axial and lateral resolution without the use of pinholes. The line illumination was achieved using a low-cost anamorphic optical lens. The description of the optical system, the acquisition system (with *PIC* microcontroller and *CMOS LISI024* sensor as the core of the printing circuit), the visualisation system and software is also included.

The preliminary results presented in these work were attained using plane mirror object mounted in a translation stage. *Matlab* routines to estimate the axial resolution and the system confocality were used.

Agradecimentos

Sem menosprezar o apoio de todos os amigos, colegas e nomeadamente da família gostaria de dirigir um agradecimento muito especial ao meu orientador Professor Doutor Carlos Correia e ao co-orientador Engenheiro Milton Macedo pelo desafio inicial proposto, por todo o apoio, paciência singular, presença constante e confiança depositados neste projecto sem os quais a realização deste não teria sido possível.

Resumo	1
Abstract	2
Agradecimentos	3
Índice	4
Lista de Acrónimos	7
1. Introdução	9
2. Microscopia confocal: sua descrição e caracterização	11
2.1. Breve história da microscopia.....	11
2.2. Microscópio convencional versus confocal	13
2.3. Descrição do método confocal.....	15
2.4. Fundamentos da formação da imagem óptica.....	19
2.5. Discriminação em profundidade (seccionamento óptico).....	21
2.6. Formas de visualização das imagens tridimensionais.....	23
2.7. Parâmetros de avaliação da qualidade da imagem.....	26
2.7.1. Resolução axial e lateral	26
2.7.2. Contraste e relação sinal/ruído.....	30
3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional	33
3.1. Introdução	33
3.2. Tipos de iluminação	33
3.2.1. Fontes de luz incoerente.....	34
3.2.2. Lasers	35
3.2.2.1. Tipos de lasers.....	36
3.3. Tipos de detectores de luz visível	39
3.3.1. Tubos fotomultiplicadores (<i>PMT</i>).....	40
3.3.2. Fotodíodo	41
3.3.3. Fotodíodo de avalanche	42
3.3.4. Sensor CCD	43

3.3.5. Sensor CMOS	46
3.3.6. CCD versus CMOS.....	47
3.4. Técnicas de microscopia tridimensional.....	49
3.4.1 Microscopia confocal de fluorescência.....	49
3.4.2. <i>PAM</i>	50
3.4.3. Microscópio confocal de linha.....	52
3.4.4. Microscópio de campo amplo com seccionamento óptico	54
3.4.4.1. Luz estruturada num microscópio convencional	54
3.4.5. Microscópio de varrimento <i>Tandem</i> (com disco de <i>Nipkow</i>)	56
4. Módulo de leitura de dados do sensor.....	58
4.1. Introdução	58
4.2. Arquitectura do módulo	59
4.2.1. Sensor <i>CMOS LIS-1024</i>	62
4.2.2. PIC16F876 TM	62
4.2.3. Amplificador TL081CPN	67
4.2.4. Protocolo RS232	67
4.2.4.1. MAX242	69
4.3. Programação do PIC	70
4.4. Programa de visualização dos dados no PC.....	74
5. Resultados e Discussão	77
5.1. Introdução	77
5.2. Montagem óptica	77
5.3. Avaliação da resposta axial do sistema.....	81
5.3.1. Lente objectiva esférica versus acromática.....	81
5.3.2. Iluminação em linha versus <i>pinhole</i> de detecção.....	83
5.3.3. Pinhole de detecção: 10 μm versus 100 μm	84
5.3.4. Efeito da utilização de um <i>LSD</i>	85
5.3.5. Filtros de polarização versus filtros <i>ND</i>	86
5.3.6. Fenda com 530 μm versus 540 μm	86

6. Conclusões e Trabalho Futuro.....	88
6.1. Conclusões	88
6.2. Trabalho Futuro	91
Referências.....	93
Apêndice A.....	98
Apêndice B.....	99
Apêndice C.....	100
Apêndice D.....	103
Apêndice E.....	110
Apêndice F	120

3D – Três dimensões
 λ – Comprimento de onda
A/D ou ADC – Analogic to digital convertor
APD – Fotodíodo de avalanche
BS – Beamsplitter (espelho dicróico ou um divisor de feixe)
CCD – Charge coupled device
CL – Microscópio confocal de linha
CLK – Clock
CLSM – Confocal laser scanning microscope
CMOS – Coplementary metal oxide semicondutores
CP – Ponto focal corrente
CPU – Central processing unit
CSOM – Varrimento confocal em tempo real
DIN – Deutsches Institut für Normung (Instituto Alemão para a Normalização)
DMD – Dispositivo de microespelhos digital
DPR – Dynamic pixel reset
DPSS – Lasers de estado sólido de bomba de diodo
FWHM – Full Width Half Maximum
ISO – International Standards Organization
LG – Line generator
LSD – Light shaping diffuser
MC – Microscópio confocal
NA – Abertura numérica
ND – Neutral density
PAM – Programmable array microscopy
PC – Personal computer
Pinhole – Diafragmas
PCB – Printed circuit board
PIC – Peripheral interface controller
Pixel – Elemento da imagem
PMT – Tubos fotomultiplicadores

PSF – Função espalhamento do ponto

RLC – Resolução lateral confocal

SLM – Modelador espacial de luz

SNR ou S/N – Razão sinal ruído

SOM – Microscópio óptico de varrimento

SPDA – Single photon avalanche diode

TAC – Tomografia axial computadorizada

TSOM – Tandem scanning optical microscope

1. Introdução

Ao longo da História o homem teve a necessidade de observar e explicar o mundo que o rodeia a uma escala inferior à vista humana^[1,3,4]. O processamento óptico de imagens, ou o sinal obtido pela percepção de objectos a essa escala, teve o início no microscópio dito normal. Neste microscópio os pontos situados fora do plano focal aparecem desfocados mas contribuem sempre para a formação da imagem.

No microscópio confocal (*MC*), inventado por *Webb*^[29] em 1981, o feixe iluminante é focado através de um sistema conjugado de furos circulares (*pinholes*) que produzem o efeito confocal que consiste no bloqueio da luz proveniente dos planos fora do plano focal. Desta maneira o observador obtém uma imagem que contém apenas pontos pertencentes ao plano focado eliminando todos os outros. A sua capacidade de formar imagens a três dimensões (*3D*) é uma grande vantagem em relação ao microscópio normal, realizando vários varrimentos a diferentes profundidades. Mas este microscópio, apesar de ser utilizado em diversas áreas com ênfase nas Ciências da Vida, e de ser uma ferramenta poderosa, é bastante caro e tem problemas ao nível da velocidade de aquisição de imagens.

Assim motivados pela necessidade de construir um novo circuito óptico, juntamente com uma placa de aquisição de dados com um sensor linear para testar um novo método de microscopia confocal, de modo a eliminar a dependência do uso dos pinholes, mas tendo um circuito mais dependente das características de um detector de luz (sensor *CCD*) e das técnicas de processamento digital de sinais subsequentes á aquisição da imagem surgiu este trabalho. O novo microscópio será testado numa mesa óptica que no início terá como objectivo testar a exequibilidade do conceito de confocalidade e em seguida, numa segunda fase, são introduzidos no circuito óptico os movimentos de varrimento necessários á formação da imagem. Espera-se como resultado final uma melhoria dos parâmetros gerais em relação ao *MC*, nomeadamente, uma significativa redução do tempo necessário para se obterem imagens a *3D* de um objecto bem como uma possível melhoria das resoluções lineares da imagem do objecto e a diminuição do ruído.

No capítulo 2 é descrita uma breve introdução histórica ao microscópio, convencional e confocal, sua descrição e caracterização. Fez-se uma descrição do método confocal, fundamentos de formação da imagem óptica do seccionamento óptico a três dimensões e dos parâmetros de avaliação da qualidade da imagem.

No capítulo 3 são descritas as diferentes fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional, assim como os tipos de iluminação e os tipos de detectores de luz visível utilizados.

No capítulo 4 descreve-se o módulo de leitura de dados do sensor, as suas características e funcionalidades. Aqui são dadas as razões para a escolha da arquitectura do módulo e os seus componentes como o sensor *CMOS LIS-1024*. A programação do *PIC 16F876* e o programa de visualização dos dados no *PC* também têm aqui referência.

No capítulo 5 é realizada a síntese dos resultados assim como a sua discussão geral com base em diferentes montagens ópticas e tendo em conta os componentes utilizados (lente objectiva ou acromática, iluminação em linha ou utilização de um *pinhole*, o efeito da utilização de um *LSD* ou de uma fenda com base na avaliação da resposta axial do sistema).

Por fim no capítulo 6 afiguram-se as conclusões deste trabalho e são identificados alguns pontos a desenvolver num trabalho futuro.

2. Microscopia confocal: sua descrição e caracterização

2.1. Breve história da microscopia

Desde os primórdios, o homem teve necessidade de observar o que o rodeava. Inicialmente, observou objectos a grande distância, resultando no Sec. X o aparecimento das lentes de ampliação. No Sec. XIII apareceram os óculos, cujo uso se tornou comum no Sec. XVI. Ainda no Sec. XVI, apareceu o telescópio composto por óptica muito simples, permitindo já uma ampliação de imagens, se bem que muito reduzida, muito usado na astronomia e na observação militar. Só nos últimos séculos da nossa história, com o desenvolvimento da óptica, cujos avanços se registaram no Sec. XIX e XX, e mais tarde da electrónica, é que o homem teve a capacidade de explicar o mundo e os fenómenos que o rodeavam a uma escala inferior à vista humana^[1,3,4].

Desde o aparecimento do primeiro microscópio até hoje houve um longo caminho a percorrer. Façamos então uma curta viagem no tempo.

O primeiro microscópio composto apareceu em 1590 pelas mãos dos holandeses, pai e filho, Hans & Zacharias Jansen. Este microscópio era limitado uma vez que ampliava apenas cerca de 20X e só podia observar objectos opacos^[4]. Marcello Malpighi, professor de medicina e anatomista, foi um dos maiores microscopistas da sua época, observando pela primeira vez as cavidades existentes na madeira e o desenvolvimento de sementes em 1660. Em 1665, Robert Hooke publicou um livro de micrografia, '*Micrographia*', descrevendo os componentes do microscópio e as observações mais importantes realizadas pelo próprio em Londres^[1,4]. Antoni Van Leeuwenhoek, designado o pai do microscópio, criou em 1673, em Delft na Holanda, um microscópio que podia fazer ampliações até 300X, publicando assim desenhos de alguns microorganismos observados como bactérias, células de sangue, sémen etc.^[1,3,4].

Só no ano de 1839 é que a unidade básica da vida, a célula, foi finalmente conhecida^[4]. Podemos ver na figura 2.1 os nomes mais importantes que contribuíram para o desenvolvimento do microscópio até ao início do século XX.

No início do século XX, o professor Köhler desenvolveu um método de iluminação designado ‘*Iluminação Köhler*’, baseado na fórmula de Abbe^[6]. Descobriu que, ao usar uma luz com comprimento de onda (λ) pequeno, dentro da gama de radiação ultra violeta, melhorava substancialmente a resolução, criando, deste modo, um campo iluminado uniforme de visão, enquanto a amostra era iluminada com um cone de luz bastante extenso^[1,2,5].

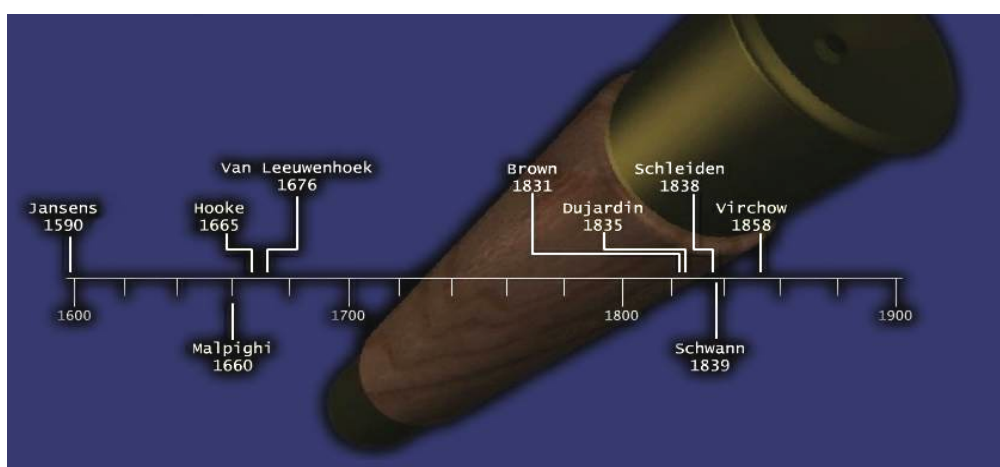


Fig. 2.1 – História do microscópio até ao início do século XX.

O microscópio, designado básico ou convencional, teve o seu aparecimento nas primeiras décadas do século XX. Originalmente ajustado pelas normas *DIN* germânicas, estes microscópios têm as distâncias ópticas e construtivas normalizadas na figura 2.2^[2].

Com base no microscópio convencional, apareceram outros tipos de derivação da estrutura inicial como o microscópio com fim perpendicular e o microscópio invertido. Surgiu posteriormente o microscópio de electrões, o microscópio fotónico, o microscópio de fluorescência e finalmente o microscópio confocal, comercializado em 1983^[1,2].

O microscópio confocal com varrimento¹ a laser (*CLSM – confocal laser scanning microscope*) foi concebido na década de 50, sendo este princípio descrito por

¹ Varrimento é o termo usado para a iluminação ou observação sequencial de áreas pequenas.

M. Minsky^[8]. Foi a partir dos anos 80 que se tornou um instrumento prático utilizado como técnica de obtenção de imagens de luz tendo sido Petran e Baer^[20, 21] os primeiros a descrever a forma de varrimento duplo. A motivação para o trabalho destes autores foi a necessidade de obter uma imagem a partir de uma determinada secção de um objecto fino translúcido, sem a presença de informação de focagem externa dos planos vizinhos. O princípio confocal preenche este requisito ao mesmo tempo que fornece uma melhor resolução lateral. Desde então tem tido muita aplicação no campo das ciências biológicas graças ao desenvolvimento do varrimento digital, dos lasers e das tecnologias de imagem^[2]. Mas estas razões só por si não são suficientes para o aumento da sua utilização. Pode-se dizer que se estes instrumentos não operassem no modo confocal não teriam tanta popularidade^[14].

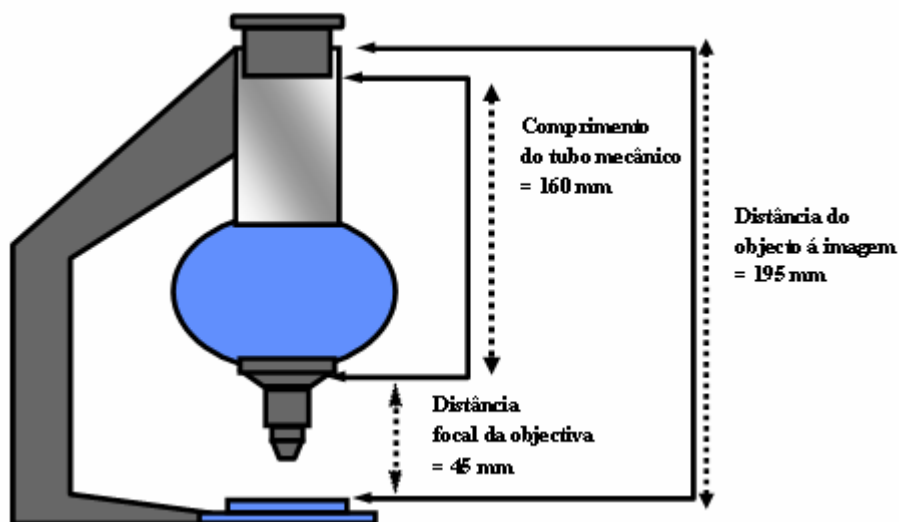


Fig. 2.2 – Microscópio convencional segundo a especificação ISO.

2.2. Microscópio convencional versus confocal

Quando se utiliza um microscópio óptico comum para observar objectos, tais como células completas, a imagem visualizada é pouco definida e todas as estruturas da célula são observadas em simultâneo. Focando uma determinada zona, também se visualizam as áreas situadas fora do foco, tendo como resultado uma imagem pouco nítida. Além disso, não permite fazer uma ideia da distribuição tridimensional das diferentes partes da amostra. Quando se querem observar imagens definidas, devem-se

efectuar cortes finos na amostra, no entanto, isto torna-se inexecuível quando se trata do estudo de células vivas. Mesmo quando se fazem cortes histológicos muito finos, e estes são observados através de um microscópio comum, não é possível observar toda a espessura do corte. Na realidade, através do ajuste do botão micrométrico pode-se variar a focagem. Mas ao focar-se num dado plano, desfocam-se os outros planos. O microscópio comum tem o inconveniente de haver sobreposição das imagens desfocadas de outros planos à imagem nítida proveniente do plano focado^[7].

A grande vantagem do *CLSM* para os biólogos é sua capacidade de produzir secções ópticas da amostra a três dimensões, ou seja, de toda a célula ou de um tecido, que contém informação mais do que de um único plano focal. O microscópio confocal soluciona o inconveniente do microscópio convencional que é o de sobrepor imagens desfocadas de outros planos à imagem nítida proveniente do plano focado^[1]. Este sistema permite apenas a observação de um plano isolado, situado no ponto do foco do sistema óptico, eliminando de forma óptica e através de um diafragma, o 'pinhole', a luz proveniente dos planos que estão fora do foco^[9,11]. Estes microscópios podem funcionar no modo de fluorescência ou no modo de campos luminosos² ou reflexão^[2, 13, 14].

Na figura 2.3 podem-se ver as diferenças entre o microscópio confocal e o convencional de fluorescência. Ambos têm como base de funcionamento o fenómeno de fluorescência. Ao fazer incidir luz numa determinada amostra, pode-se ver luz de diferentes cores emitidas a partir da mesma^[2]. O microscópio confocal de fluorescência combina o microscópio de fluorescência com a análise electrónica da imagem, proporcionando imagens a três dimensões, o que não se verificava noutra tipo de microscópios^[13].

Para o modo de reflexão^[13] é usado um espelho semitransparente que reflecte 50% da luz laser incidente, através da lente objectiva, para reagir com a amostra. A luz é então reflectida pela amostra e esta é colectada pelo detector. Para suprimir a luz reflectida pelos diversos elementos ópticos do microscópio, um prato com $\lambda/4$ e um filtro polarizado são colocados no caminho da luz^[13].

As vantagens que se obtém com a utilização do microscópio confocal são:

² Tradução adoptada para *Brightfield Microscope*

- Redução das manchas da imagem a partir da luz de varrimento;
- Melhoria da relação sinal ruído;
- Visualização limpa de um objecto fino (melhor contraste);
- Seccionamento óptico através de varrimentos no eixo dos Z (eixo óptico);
- Melhor resolução das imagens multidimensionais;
- Ajuste electrónico da ampliação ^[3].

Consequentemente o *CLSM* tornou-se uma ferramenta importantíssima para a investigação nos campos da biologia, medicina ou ciência dos materiais.

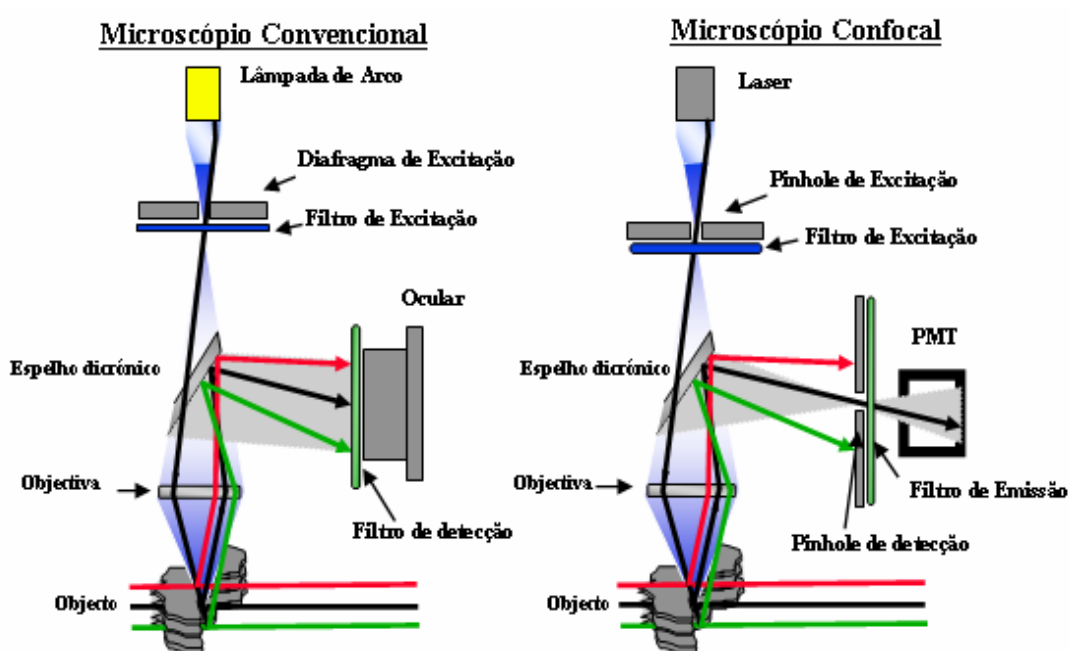


Fig. 2.3 – Comparação entre microscópio convencional e confocal de fluorescência

2.3. Descrição do método confocal

Tendo como referência o esquema da figura 2.4, o objecto é iluminado por uma luz, que tem origem na fonte laser externa, via uma lente objectiva. A fonte é pontual, ou seja, é utilizada para atingir um único ponto no objecto. A resolução depende, em maior grau, da lente objectiva que vai percorrer o objecto ponto por ponto, com um ponto de luz, ou mancha de luz focado. A segunda lente serve apenas para recolher a informação vinda da luz transmitida pelo objecto^[15,16]. O varrimento de

detecção de um ponto no plano da imagem para medir a sua intensidade é simples. Para que a segunda lente contribua para a resolução, é necessário que essa lente se torne numa lente de imagem, mais do que uma lente colectora. Isso é conseguido utilizando um detector para cada ponto, não sendo necessário uma detecção da área total^[16]. Esta é a forma como o microscópio confocal funciona. Aqui a luz vinda de uma pequena parte do objecto, que é iluminada pela fonte, é detectada ponto por ponto pelo detector. Desta forma, pode-se construir uma imagem, ponto por ponto, que será posteriormente armazenada e tratada num computador e visualizada num ecrã. Este esquema também é usado como sistema em modo de reflexão^[15].

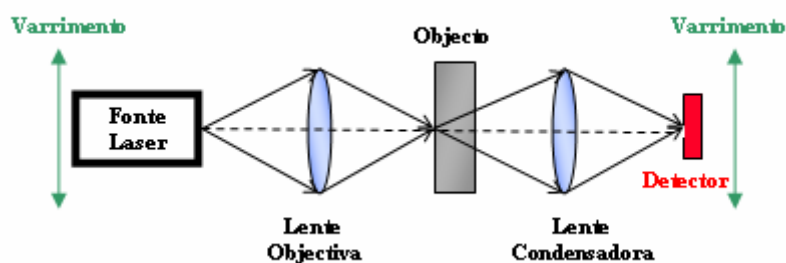


Fig. 2.4 – Esquema do processo de formação de imagem no microscópio confocal em modo de transmissão.

Desta configuração confocal simétrica, observa-se que ambas as lentes desempenham papéis semelhantes na imagem e a resolução vai deste modo aumentar^[14]. O uso do detector faz com que a segunda lente possa focar a região do objecto de prova focado pela primeira lente. Inicialmente esta melhoria pode não parecer plausível, contudo, pode ser explicado por um princípio dado por Lukosz (1966) que refere na sua essência que a resolução pode ser aumentada com a expansão do campo de visão^[17]. Uma maneira de melhorar com o princípio de Lukosz é colocar uma abertura extremamente pequena perto do objecto^[18,19]. A resolução é agora determinada pelo tamanho da abertura e não pela radiação e o campo de visão é aumentado pelo varrimento. No microscópio confocal, não se utiliza uma abertura física no plano focal usa-se uma imagem projectada por trás de um ponto do detector juntamente com um ponto da fonte focado^[18,19].

Observando agora o diagrama de um microscópio confocal representado na figura 2.3, vê-se que a iluminação é realizada por um feixe de luz laser que após passar

num *pinhole*, e através de um varrimento ilumina ponto por ponto apenas um determinado plano da amostra, realizando um verdadeiro ‘corte óptico’^[12,13]. A fonte de luz laser polarizada e colimada é desviada em pequenos passos discretos nas direcções laterais definidas pelos eixos X e dos Y por uma unidade de varrimento, mas antes passa através do espelho dicróico ou um divisor de feixe (*beamsplitter*) que a faz incidir na amostra depois de focada pela lente objectiva do microscópio^[13]. A luz vinda da amostra atravessa a lente objectiva no sentido inverso, é reflectida por um espelho dicróico, e passa através do *pinhole* colocado antes do detector^[13, 14].

A imagem é obtida usando um arranjo de dois diafragmas (*pinholes*) localizados em pontos conjugados do percurso óptico, que agem como fonte de luz pontual e detector pontual^[12]. Os raios luminosos que estão fora de foco são eliminados pelo *pinhole* de detecção. No esquema da figura 2.3, pode-se observar um objecto tridimensional que, conseqüentemente, contém vários planos focais. A luz de todos os planos é refractada pela lente, mas só a proveniente do plano central, do foco, atravessa o *pinhole*, formando assim uma imagem nítida e perfeitamente definida das estruturas situadas no plano focal, sem interferência da luz proveniente do resto do objecto^[13]. Para estudar um objecto em microscopia confocal usam-se imagens obtidas para os diferentes planos focais.

Cada uma destas imagens designa-se por secção óptica porque corresponde à imagem obtida ao seccionar a amostra com um micrótomo (seccionamento óptico). Como se pode ver na figura 2.5. o seccionamento óptico é feito segundo o eixo dos Z, enquanto o varrimento da luz laser é realizado em cada secção no plano XOY.

Assim a imagem de cada plano é formada apenas pelas estruturas que se encontram no plano de varrimento^[12]. O sistema de varrimento permite que a estimulação não seja feita de uma só vez em toda a amostra mas realizada por etapas em áreas muito pequenas. A área de estimulação (e de recepção) vai-se movendo em toda a amostra³ de forma que, num dado intervalo de tempo de varrimento, cubra a totalidade da amostra. Não somente a imagem é mais nítida, como também a amostra pode ser cortada opticamente em vários planos^[12,13]. As imagens assim obtidas dos ‘cortes ópticos’ podem ser armazenadas em suporte digital (discos rígidos, CD-ROM,

³ Existem diferentes formas de varrimento como um feixe estacionário e um objecto em movimento.

etc.) e utilizadas para, com recurso a um computador, construir uma imagem tridimensional ou efectuar cálculos do comprimento, área, volume e outras análises de acordo com a finalidade do estudo^[10].

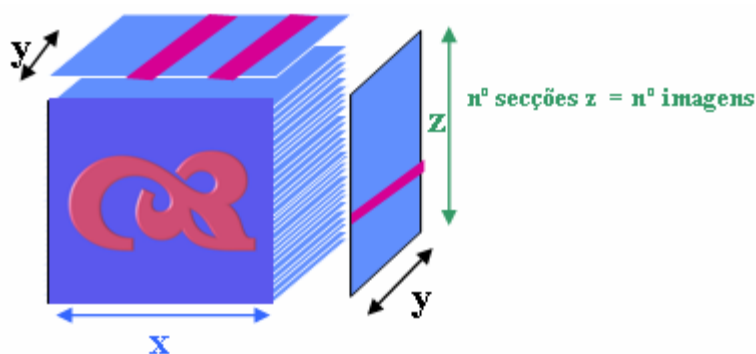


Fig. 2.5 – Seccionamento óptico do microscópio confocal.

Para obter a imagem completa, o ponto de luz ilumina a amostra num movimento de varrimento, controlado por espelhos deflectores. A luz emitida ou reflectida pela amostra, que passa pelo *pinhole* de detecção, é transformada em sinais eléctricos por um fotomultiplicador ou outro tipo de detector, sendo visualizada no monitor do computador. Alterando passo a passo o plano de foco, obtém-se uma pilha de imagens (secções ópticas) e, com recurso a sistemas informáticos adequados, pode-se realizar uma reconstrução tridimensional do objecto, do mesmo modo que, com os diferentes cortes de um sistema radiológico de *TAC* ou ressonância magnética, podem reconstruir-se informaticamente estruturas anatómicas humanas^[13].

Em síntese podem-se estabelecer os seguintes requisitos para obter uma boa qualidade de cada imagem focal e consequentemente de reconstrução tridimensional:

- Resolução axial – devem-se obter bons valores da FWHM (largura a meia altura) da curva de intensidade para uma secção vertical de imagens da amostra;
- Campos planos – devem ser capazes de colectar imagens de um campo plano de uma amostra ou a informação do eixo dos *Z* é incorrecta;
- Aberrações cromáticas – deve-se testar que a emissão de radiação é constante e não são colectados os efeitos tangenciais ou radiais impostos pelas aberrações cromáticas nas objectivas;

- Exactidão e precisão na condução de Z – deve ser capaz de reproduzir detalhes na amostra com precisão na ordem do micrómetro que poderá alterar a imagem final^[1,3].

2.4. Fundamentos da formação da imagem óptica

Por não ser do âmbito deste trabalho não se vão apresentar grandes detalhes da teoria da formação da imagem nos microscópios convencionais e confocais. Contudo, é importante apresentar alguns aspectos teóricos de modo a perceber melhor as diferenças entre diferentes tipos de microscópios. Será usada a aproximação paraxial e será simplificada pelo facto de se considerarem apenas amostras finas, caracterizadas por uma função de reflexão ou transmitância complexa t ^[14, 27, 28]. A função é complexa de forma incluir as variações de fase causadas pelas variações de espessura ou rugosidade de superfície. Começa-se por considerar a formação da imagem por uma lente simples, como mostra a figura 2.6. Sendo a amplitude do campo no plano (x_1, y_1) dada por $U(x_1, y_1)$, então pode-se escrever o campo no plano (x_3, y_3) , desprezando constantes premultiplicativas e variações de fase, como^[27, 28]:

$$U(x_3, y_3) = \iint U(x_1, y_1) h(x_1 + x_3 / M, y_1 + Y_3 / M) dx_1 dy_1 \quad (2.1)$$

M – Amplificação que tem o valor de d_2/d_1 ;

h – São os pontos de amplitude da função de expansão (ou resposta ao impulso) que é dada por^[28]:

$$h(x, y) = \iint P(x_2, y_2) \exp \frac{2\pi j}{\lambda} (x_2 x + y_2 y) dx_2 dy_2 \quad (2.2)$$

P – Função pupila da lente.

A função pupila serve para descrever a extensão física da lente e a sua reflexão. É igualmente uma função complexa que descreve as aberrações presentes, incluindo o nível de desfocagem do sistema.

Supõe-se agora que $U(x_1, y_1)$ representa o objecto que consiste simplesmente num único ponto brilhante num fundo opaco, por isso:

$$U(x_1, y_1) = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \quad (2.3)$$

$\delta()$ – Representa a função delta de *Dirac*.

A intensidade na imagem é dada pela equação 2.1:

$$I(x_3, y_3) = |h(x_3 / M, y_3 / M)|^2 \quad (2.4)$$

Considerando uma função pupila circular simétrica de raio a , pode-se introduzir uma função pupila $P(\rho)$ onde $\rho = r_2 / a$ que é zero para $\rho \geq 1$. Agora na direcção correcta pode-se escrever a equação como^[15, 16]:

$$I(v) = \left| 2 \int_0^1 P(\rho) J_0(v \rho) \rho d\rho \right|^2 \quad (2.5)$$

J_0 – Função de *Bessel* de ordem zero;

v – Coordenada óptica normalizada relativamente à coordenada real radial, r_3 ,

e o seu valor é determinado por $v = \frac{2\pi}{\lambda} r_3 \sin \alpha$:

$\sin \alpha$ – A abertura numérica (*NA*) da lente (sendo o meio envolvente o ar) que é dada por a / d_2 .

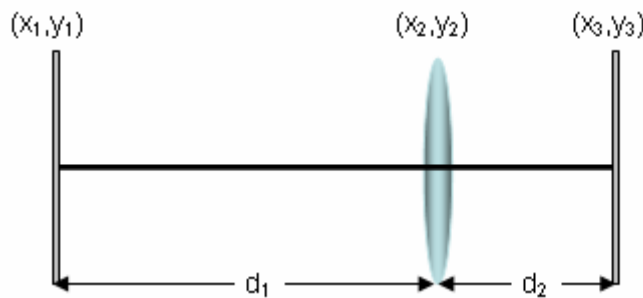


Fig. 2.6 – Geometria de formação da imagem.

É claro que a forma da imagem agora depende da forma da função pupila, $P(\rho)$. No caso ideal:

$$p(\rho) = \begin{cases} 1 & \rho < 1 \\ 0 & \text{para os outros casos} \end{cases}$$

Pode-se escrever:

$$I(v) = \left(\frac{2J_1(v)}{v} \right)^2 \quad (2.6)$$

que tem a forma normal do disco de *Airy*^[14]. O primeiro zero ocorre quando $v = 1,22\pi$.

Como exemplo limite da forma da função pupila, considera-se uma pupila em forma de anel infinitamente fina de modo a que a luz seja transmitida só no raio normalizado $v = 1$. A partir da equação 2.5 obtém-se:

$$I(v) = J_0^2(v) \quad (2.7)$$

Deve-se recordar que na equação 2.7 foi desprezado o termo de fase premultiplicativa. Deve-se ao facto de não ser importante nesta discussão da imagem de um ponto do objecto, mas será para a imagem de objectos estendidos a maiores aberturas ou baixos valores do *número de Fresnel*^[27]. O *número de Fresnel* é um parâmetro adimensional e é definido como $F = \frac{a^2}{R\lambda}$, onde a é o raio da abertura, λ é o comprimento de onda da radiação e R é a distância focal até á abertura. Quando $F < 1$ o regime é designado de difracção de *Fraunhofer*, e quando $F \geq 1$ produz-se o difracção de *Fresnel*. Os regimes intermédios requerem uma análise mais difícil, mas podem ser tratados usando a teoria de difracção escalar^[27, 32].

2.5. Discriminação em profundidade (seccionamento óptico)

A luz, vinda da amostra focada por uma abertura fina, assegura que a informação obtida é só de um nível da amostra pretendido. O seccionamento óptico ou propriedade de discriminação em profundidade tornou-se na maior motivação para a utilização dos microscópios confocais e é a base de muitos dos novos métodos de obtenção de imagem.

Na figura 2.7 mostra-se o modo de reflexão do microscópio confocal. As linhas contínuas mostram o caminho óptico quando as saliências do objecto permanecem no plano focal da lente. Até à posição final de varrimento, a superfície do objecto é suposta estar localizada no plano vertical das linhas a tracejado. Neste caso, o traçado de um raio simples, mostra que a luz que reflecte para trás do detector chega como uma mancha desfocada. Só a porção central que é determinada, vai contribuir para a imagem. Desta maneira, o sistema discrimina as características que não

permanecem na região focal da lente. Um método muito simples que varre numa reflexão axial perfeita através do foco e mede a intensidade do sinal detectado^[14].

A figura 2.8 mostra uma resposta típica de um sensor, ou detector, em função da posição axial de uma superfície planar reflectora, que é frequentemente designada de $V(z)$, por analogia com técnicas similares na microscopia acústica de varrimento^[22]. Pode-se descrever a distribuição de intensidade na imagem assim obtida, para um objecto plano, na aproximação paraxial, através da expressão:

$$I(u) = \frac{\sin\left(\frac{u}{2}\right)^2}{u/2} \quad (2.8)$$

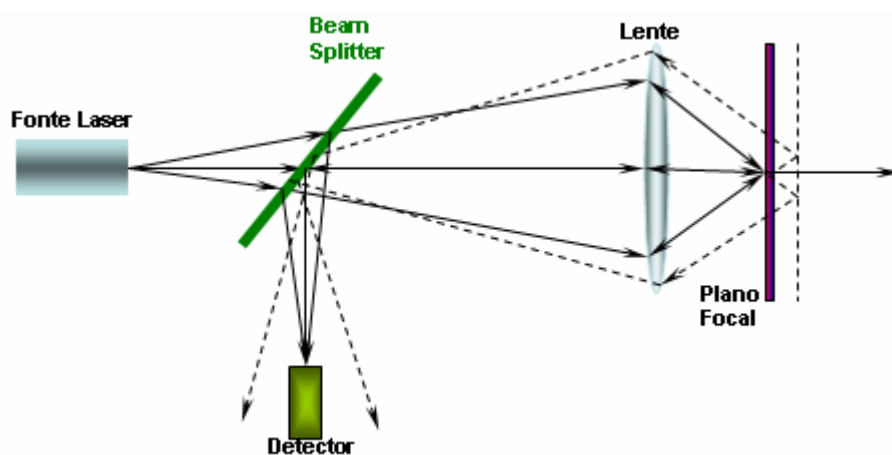


Fig. 2.7 – Esquema do processo de formação de imagem no microscópio confocal em modo de reflexão.

A coordenada normalizada axial, u , é obtida a partir da distância axial real z , sendo determinada por:

$$u = \frac{8\pi}{\lambda} z \sin^2\left(\frac{\alpha}{2}\right) \quad (2.9)$$

Sendo:

λ – Comprimento de onda da luz laser;

α – Semi-ângulo definido pelo cone de luz incidente na objectiva

$\sin \alpha$ – Abertura numérica da objectiva se o meio envolvente é o ar (pois a expressão geral é $n \sin \alpha$ e $n_{ar}=1$).

Considerando outras características do objecto, como pontos e linhas, podem-se obter diferentes graus de seccionamento^[14]. Em particular, a resposta para um ponto do objecto é dada por:

$$I(u) = \frac{\sin\left(\frac{u}{4}\right)^4}{\frac{u}{4}} \quad (2.10)$$

Mostrando assim que o seccionamento é mais fraco que no caso anterior para um plano.

Embora o grau de seccionamento óptico seja dependente da forma do objecto, uma superfície planar reflectora prova ser um objecto muito útil para testar o alinhamento e desempenho de um microscópio^[14].

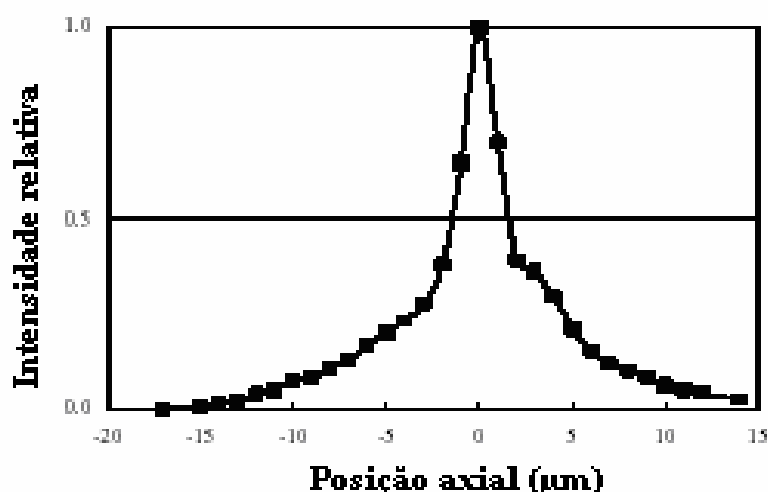


Fig. 2.8 – Resposta típica do sensor a um varrimento axial^[22].

2.6. Formas de visualização das imagens tridimensionais

Esta propriedade, discriminação em profundidade, é a maior razão para a popularidade do microscópio confocal. A capacidade de seccionamento é mais forte do que a focagem em profundidade o que sugere que detalhes de uma região volumétrica específica de uma amostra fina translúcida possam ser captados através da escolha da

posição focal^[14]. Em resumo, pode-se seccionar uma amostra opticamente sem ter de usar meios mecânicos.

A figura 2.9 mostra um esquema idealizado do processo. A porção do objecto intermédia que se quer ver, em destaque na referida figura, é determinada pela posição do foco. Desta maneira, é possível retirar, através de uma série de focos, os dados relativos a estruturas a três dimensões da amostra. Representando a imagem tridimensional por $I(x,y,z)$ então através da focagem na posição $z=z_I$, obtém-se idealmente, a imagem $I(x,y,z_I)$ ^[14, 23]. Na prática, não corresponde à verdade porque a secção óptica não é infinitamente fina.

O microscópio confocal fornece imagens que se formam com grande resolução e com uma profundidade de focagem suficientemente pequena para que todos os detalhes que são adquiridos apareçam no foco. Isto sugere imediatamente que se pode estender a profundidade de focagem do microscópio por adição do conjunto ou integração, das imagens retiradas de diferentes posições do foco sem sacrificio de resolução lateral^[14]. Matematicamente, estas imagens do foco estendido são dadas por:

$$I_{FE}(x,y) = \int I(x,y,z)dz \quad (2.11)$$

Como alternativa ao método de focagem estendida, pode-se formar uma imagem auto-focada por varrimento da linha do eixo do objecto e, em vez de integrar, selecciona-se o foco para cada ponto da imagem, gravando o máximo do sinal detectado^[23]. Matematicamente, isto pode ser escrito por:

$$I_{AF}(x,y) = \int I(x,y,z_{max})dz \quad (2.12)$$

z_{max} – corresponde à posição do foco dando o sinal máximo.

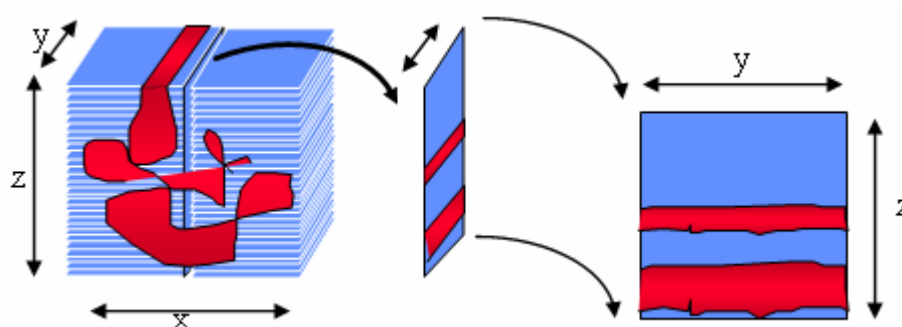


Fig. 2.9 – Esquema da discriminação em profundidade.

A imagem obtida é de certa forma similar ao foco estendido e pode ser obtido um aumento substancial na profundidade do foco.

A sensibilidade desta técnica de perfil pode ser melhorada, recorrendo a técnicas de interferência^[24]. A ideia base consiste em mover o objecto na direcção axial, de um ponto para outro, desde que o interferómetro esteja sempre fechado para uma franja negra^[14]. Em vez de mover o objecto axialmente, podemos incorporar a mudança de fase no ramo de referência do interferómetro. Esta é a vantagem para grandes velocidades e uma sensibilidade e resolução melhor do que 1nm .

O método confocal fornece uma ferramenta própria conveniente para estudar estruturas a três dimensões. Recolhe-se a imagem com uma série de slides e enviámo-los de volta na forma desejada. Naturalmente na prática não é tão simples como parece, mas pode-se, por exemplo, mostrar os dados como uma imagem nos eixos x - z mais do que uma imagem x - y . Isto é de alguma forma similar a uma vista de lado da amostra. Como outro exemplo, poder-se-á escolher a recombinação de dados como um par estéreo por introdução insignificante de desvio lateral para cada fatia da imagem a fim de serem adicionados depois conjuntamente^[25]. Ao fazer isto duas vezes, com um desvio para a sua esquerda num caso e para a direita no outro caso, obtém-se simplesmente um par estéreo. Matematicamente, chega-se à seguinte forma:

$$\int I(x \pm \gamma, y, z) dz \quad (2.13)$$

γ – É uma constante.

Na prática, pode não ser necessário introduzir desvios em ambas as direcções para obter uma vista estéreo adequada.

O ponto-chave é que tanto no modo de campo brilhante como no de fluorescência o princípio confocal permite que a imagem da amostra seja obtida a três dimensões. O conhecimento completo do processo de formação da imagem, juntamente com o efeito de aberração das lentes e a absorção, é necessário antes da correcta manipulação dos dados^[26]. Tudo o que é necessário é uma boa fonte laser, uma lente objectiva e um detector.

2.7. Parâmetros de avaliação da qualidade da imagem

Para se obter uma imagem de qualidade dever-se-á ter em atenção duas características da imagem que são o contraste e a resolução. A resolução é um dos conceitos que em física pode ser descrito, medido e manipulado de acordo com regras que são descritas a partir da geometria do sistema. O contraste, por outro lado, está relacionado com a inevitabilidade da existência de ruído das medidas reais que limitam a possibilidade de se usar uma resolução válida, podendo mesmo o contraste modificar a imagem real. O microscópio confocal vai aumentar o contraste e, por conseguinte a imagem exibe mais facilmente uma boa resolução. O microscópio confocal tem uma resolução melhor do que o microscópio de campo largo⁴, mas não é a única fonte do sucesso do microscópio confocal.

2.7.1. Resolução axial e lateral

O limite de resolução é definido como a menor separação entre dois objectos pontuais que podem ser distinguidos sem haver confusão ou interferência entre eles^[30]. No microscópio confocal, a resolução lateral medida em qualquer direcção num plano perpendicular ao eixo óptico distingue-se da resolução axial medida ao longo do eixo óptico. De acordo com o critério de *Rayleigh*, dois objectos pontuais estão no limite de resolução quando o centro de um dos discos de *Airy* se situa no primeiro mínimo do outro disco *Airy*^[30,31].

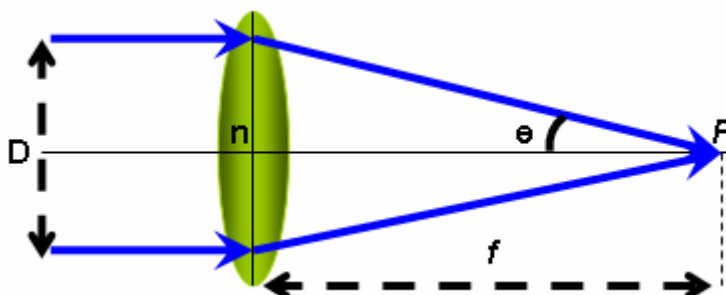


Fig. 2.10 – Representação dos parâmetros que definem a *NA* de uma lente objectiva.

⁴ Tradução adoptada para *Wide-Field Microscope*

A figura 2.10 representa alguns parâmetros que intervêm na definição de NA de uma lente objectiva. A NA da objectiva do microscópio é uma medida da sua capacidade de recolha de luz. Outras medidas são o *número de f*, que é igual a f/D , sendo f é a distância focal. Quanto maior a NA menor o *número de f*, e mais luz é colectada pela objectiva^[6,29,30].

A NA de uma lente objectiva é, então, uma medida do tamanho angular do cone de focagem da luz como se observa na figura 2.9 e é definido como^[6,29,30]:

$$NA = n \sin \theta \quad (2.14)$$

Em que:

θ – Metade do ângulo do cone;

n – Índice de refração do meio que rodeia o cone.

O raio de um disco de *Airy* é definido como a distância desde o seu centro até ao primeiro mínimo, sendo esta igual à resolução lateral de uma lente objectiva. O raio do disco de *Airy* é dado por^[6,29,30]:

$$r_{AIRY} = 0.61 \frac{\lambda}{NA} \quad (2.15)$$

Sendo:

λ – Comprimento de onda da luz;

NA – Abertura numérica da lente objectiva.

A resolução lateral do microscópio confocal também depende do campo de visão. À medida que o tamanho do *pinhole* confocal vai diminuindo, o campo de visão torna-se mais concentrado e a resolução lateral melhora com um factor de $\sqrt{2}$ ^[6,29]. Assumindo um grande nível de confocalidade, a resolução lateral confocal (RLC) pode ser escrita da seguinte maneira:

$$RLC = \frac{r_{AIRY}}{\sqrt{2}} = \frac{\lambda}{1.4} \times 0.61 \frac{\lambda}{NA} \quad (2.16)$$

A resolução axial tem um significado análogo à lateral mas em relação ao eixo óptico. A equação 2.16 dá uma expressão ambígua para resolução lateral confocal. É mais difícil obter uma expressão equivalente para resolução axial, especialmente quando se lida com fotoluminescência ou amostra de reflexão difusa. O manual de

microscopia confocal de *J. B. Pawley* confere várias expressões para a resolução axial, d_z , das quais duas se apresentam como mais relevantes ^[2,29]:

$$d_z(PL) = \frac{1.77}{(NA)^2} \quad (2.17)$$

$$d_z(RL) = FWHM = \frac{0.22}{n \sin^2(\frac{\theta}{2})} \quad (2.18)$$

Ambas as fórmulas se dizem ideais, pois considera-se que o *pinhole* tem tamanho zero. A equação 2.17 aplica o critério de *Rayleigh* ao longo do eixo ^[31]. A objectiva do microscópio focada no ar produzirá uma figura de difracção a três dimensões limitada ao ponto com um disco *Airy* de secção transversal no ponto focal no plano x-y e outra distribuição ao longo do plano x-z e y-z. A equação 2.17 descreve melhor a resolução axial em fotoluminescência em imagens latentes. A equação 2.18 é obtida recorrendo à aproximação paraxial (pequenos ângulos) assumindo que o objecto que é um espelho plano perfeito. Para pequenas aberturas numéricas, a equação 2.17 concorda com a equação 2.18 excepto por um factor de 2 devido à reflexão especular. Sublinhe-se que as equações 2.17 e 2.18 serão consideradas só aproximações que, em certas circunstâncias, irão falhar na previsão de resoluções com precisão ^[2,29].

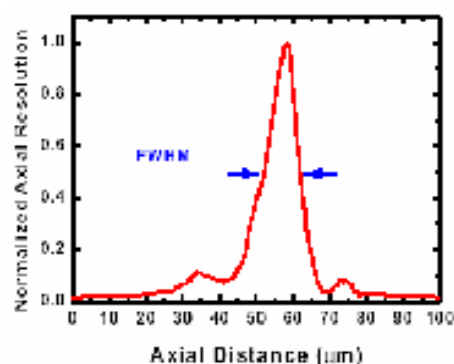


Figura 2.11 – Representação do FWHM normalmente usado como definição da resolução axial ^[29]

É importante medir experimentalmente a resolução axial e lateral para várias objectivas do microscópio, combinações de tamanho do *pinhole* e lentes colectoras de modo a caracterizar apropriadamente o microscópio confocal em questão ^[29]. A resolução lateral pode ser medida por projecção da imagem de um alvo de teste apropriado, com uma série de linhas horizontais e verticais altamente reflectoras e com

espaçamento conhecido. Uma medida de resolução axial, que está de acordo com a equação 2.18, é obtida pela projecção da imagem de um espelho plano em várias posições ao longo do eixo óptico (direcção z). A representação da resolução axial normalizada em relação à posição z num gráfico, resulta numa curva como a apresentada na figura 2.11. *FWHM* para esta distribuição é uma definição comum da resolução axial^[2,29].

Pode-se ver na figura 2.12 uma grande diferença entre as medidas experimentais da resolução axial e lateral. A resolução axial, que é verificada na figura 2.12 b), difere da resolução lateral, na figura 2.12 a), já que cada uma delas vai ser medida em eixos perpendiculares.

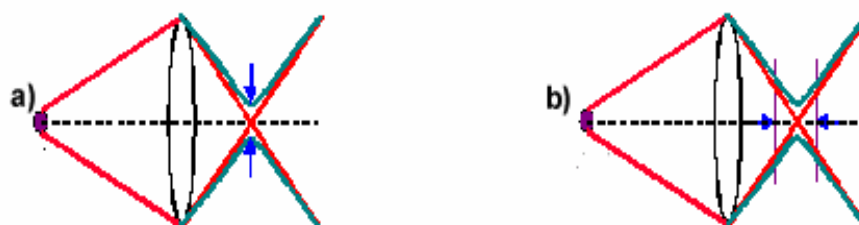


Figura 2.12 – Representação esquemática da a) resolução lateral e b) resolução axial.

Na figura 2.13 podem-se constatar as diferenças da obtenção da função espalhamento do ponto (*PSF*) em relação ao eixo axial e lateral de acordo com as coordenadas ópticas^[29].

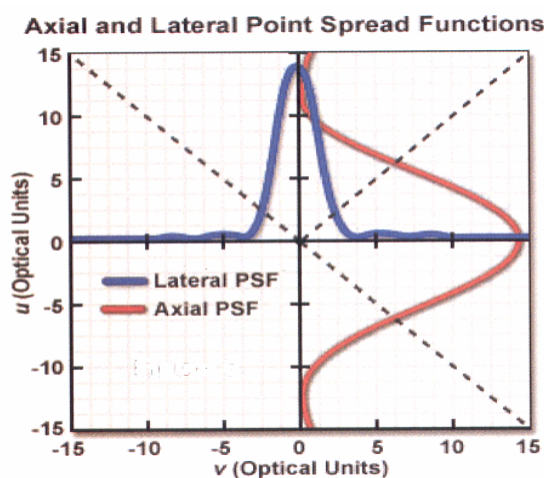


Figura 2.13 – Representação esquemática da função *PSF* na direcção axial e lateral^[29]

2.7.2. Contraste e relação sinal/ruído

Através das curvas de intensidade pode-se determinar a profundidade no vale que separa as intensidades de dois picos brilhantes consecutivos, que é necessária para distinguir dois pontos podendo ser caracterizada pela intensidade relativa desses dois picos. Designa-se por contraste de intensidade relativa, e representa-se por C , de acordo com a expressão:

$$C = (n_b - n_d) / 2n_{méd} \quad (2.19)$$

Sendo:

n_b – Intensidade das partes brilhantes (*bright*);

n_d – Intensidade das partes escuras (*dark*).

$$n_{méd} = \frac{n_b + n_d}{2}.$$

Designa-se de ‘visibilidade’ das franjas ao valor de $2n_{méd}$ ^[33]. Aumentando o tamanho das zonas de onde provêm os valores n o ruído pode ser reduzido.

Por outro lado a divisão da imagem em *pixels* que correspondem a cada medida realizada é intrínseca dos aparelhos da medida. Não podem ser confundidos com os *resels*, que resultam da óptica, sendo definidos como o elemento de resolução de uma lente que depende da sua NA . Definem-se como metade do diâmetro do disco de *Airy*, sendo portanto o raio da primeira franja escura da figura de difracção como mostra a figura 2.14.

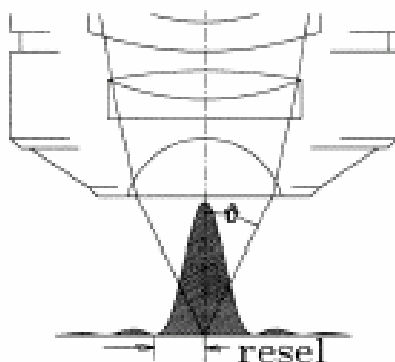


Fig. 2.14 – Representação da definição de *resel*^[33]

Para que a imagem possa reproduzir fielmente o objecto é necessário que contenha, pelo menos, dois *pixels* por *resel* em cada dimensão. Esta condição é conhecida como o *critério de Nyquist*, e apela a qualquer processo de amostragem^[29]. O requerimento da amostragem deve ser ligeiramente circular, como foi indicado pela definição do *resel*.

Assim o *pixel* precisa de ser o mais largo possível para suavizar o ruído, mas pequeno o suficiente para não degradar a resolução. No pior caso os dois pontos consecutivos, que têm diferente brilho intrínseco, sobrepõem-se e, por isso, mesmo sem ruído podem não se diferenciar.

Como se viu atrás o contraste é o termo geralmente utilizado para traduzir a visibilidade, no entanto é reconhecido que a visibilidade é determinada pelo ruído⁵.

O tempo de chegada dos fotões ao detector é aleatório, existindo então um ruído intrínseco (irreduzível) designado por ruído dos fotões, ou ruído *quantum*, que limita o sinal do detector. Este ruído é proporcional à raiz quadrada do número de fotões, isto é, o número de fotões recolhidos em cada *pixel* do detector ou em cada integração do detector ou amostragem, tem-se então^[10, 29]:

$$\text{ruído} = \sqrt{\text{número de fotões}} \quad (2.20)$$

Os valores n_d e n_b são importantes e são directamente proporcionais à tensão do sinal para cada *pixel* então:

$$v \propto n_b - n_d \quad (2.21)$$

O ruído para um dado *pixel* é:

$$N \propto \sqrt{n_{\text{méd}}} \quad (2.22)$$

Por isso a razão sinal/ruído é dado.

$$SNR = (n_b - n_d) / \sqrt{n_{\text{méd}}} = C \sqrt{n_{\text{méd}}} \quad (2.23)$$

O número de fotões depende do poder de iluminação⁶ e da eficiência do sistema óptico da colecta em partícula da lente objectiva, nomeadamente do tamanho

⁵ Para ilustrar pode-se considerar o exemplo de uma página poder ser lida com a luz do dia mas não com a luz de uma estrela. Na realidade, a impressão a preto reflecte cerca de 5% da luz que cai nele ($n_d = 0.05 n_b$) por isso o contraste nos dois casos tem o mesmo valor, $C=0.9$. No entanto há poucos fotões provenientes da luz da estrela.

⁶ Voltando ao exemplo anterior, com luz das estrelas, a página com $C=0.9$, tem $SNR \approx \sqrt{n_{\text{méd}}}$, que se torna insignificante.

angular do cone de focagem da luz (representado na figura 2.10), na medida em que determina o nível de iluminação.

O contraste é uma característica do objecto e da luz dispersa (de fundo) que é excluída no percurso óptico devido à geometria do microscópio. Por luz dispersa, entende-se a que é adicionada sem ser parte do sinal, sendo igual para todos os *pixels* vizinhos, que deverá ser reduzida dado tratar-se de uma fonte de ruído.

Existe um grande esforço em produzir microscópios em sistemas complexos para reduzir a luz de fundo (dispersa) pois assim a resolução total válida da óptica pode ser alcançada ou pelo menos mais aproximada^[35]. Por exemplo, um microscópio no modo de contraste de fase só usa luz que foi retardada em fase pela passagem através do objecto e no modo de fluorescência obtém imagem projectada que usam luz unicamente que vem de um ponto do objecto conjugado com o detector e a fonte.

Outra forma de reduzir a luz de fundo é através da realização de um varrimento pois apenas a energia dos fotões numa determinada área do feixe reage com o detector e então, o contraste não é estragado por fotões de fundo indesejados^[35]. Este tipo de realce de contraste é uma das características dos fundamentos do microscópio confocal que, com óptica especializada para maximizar o efeito, permite a observação ao longo do feixe (d direcção axial).

3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

3.1. Introdução

A imagem confocal é o exemplo mais comum de uma imagem tridimensional sendo construída num processo cuja base é a colecta sequencial de pontos individuais de luz a partir da secção volumétrica da amostra. Por isso é importante saber qual a fonte de luz a utilizar, o processo de recolha de informação e o respectivo sensor para detectar os dados retirados da imagem pela luz reflectida no objecto. Segue-se o processamento electrónico de sinal e termina com a visualização dos pontos da imagem num *display*. O processo de colecta do sinal necessita de um mecanismo de varrimento de um feixe de iluminação focado através do volume da amostra sob observação. Existem no entanto outras técnicas de microscopia tridimensional em que as fontes de luz e os sensores mais adequados variam de acordo com as suas particularidades pelo que antes de descrever essas técnicas são apresentados os tipos mais importantes das fontes de luz e sensores usados em microscopia.

3.2. Tipos de iluminação

A iluminação, ou fonte de luz, utilizada define se um sistema óptico é coerente ou incoerente. A iluminação coerente torna o sistema linear relativamente à amplitude do campo e a iluminação incoerente linear ao quadrado da amplitude da distribuição do campo de energia^[28]. Os sistemas incoerentes são mais simples de realizar, menos sensíveis ao alinhamento (contaminações de pó nos componentes). Contudo não apresentam um plano de frequências fácil de manipular como os sistemas coerentes, com os quais é possível realizar operações de filtragem pela simples manipulação de quantidades complexas resultantes da transformada de *Fourier* obtida por uma lente^[28].

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

A iluminação num microscópio confocal pode ser realizada com qualquer dispositivo que faz incidir fótons no objecto ou qualquer fonte com λ a partir do ultravioleta até ao infravermelho sendo o mais comum os *lasers* para os microscópios confocais de varrimento (*CLSM*)^[28]. Os lasers têm gama grande de radiações, são monocromáticos e altamente coerentes. Os microscópios convencionais usam geralmente iluminação incoerente. A iluminação coerente causa problemas no microscópio devido a efeitos de interferência indesejados que degradam a imagem. A luz pode incidir no objecto do lado oposto ao observador ou pode ser epitaxial (vem do lado do observador). A iluminação epitaxial é usada para observar objectos opacos e fluorescentes. Materiais opacos como os *chips* de silício têm de ser vistos por luz dispersa e reflectida a partir da sua superfície. Objectos fluorescentes aparecem com contraste elevado porque são observados com luz de maior λ do que o de iluminação, podendo esta ser rejeitada. Muitos microscópios confocais são epitaxiais sendo então uma lente objectiva utilizada duas vezes, na iluminação e detecção (observação)^[29].

3.2.1. Fontes de luz incoerente

Uma fonte de luz ideal tem uma saída igual para todas as cores (luz branca pura). A fonte de luz mais comum no microscópio de fluorescência, com elevado brilho no espectro visível e ultravioleta é a *lâmpada em arco de mercúrio*^[7,29]. O seu espectro não é contínuo, pois a luz de saída é concentrada em algumas bandas, designadas de linhas, e cada linha tem perto de 10nm de largura. Muitos sistemas propostos necessitam de filtros de excitação que transmitem uma ou mais linhas. Um filtro de excitação passa-banda de banda larga irá fornecer um sinal de emissão mais brilhante do que os filtros que excluem esta linha. Para muitas amostras é esperado uma redução no S/N total devido ao aumento de ruído da autofluorescência que irá compensar o aumento do sinal de fluorescência^[7]. Para certas aplicações envolvendo baixos níveis de fluorescência, ou para amostras com espectros na zona azul, em que um filtro de excitação passa-banda é incluído para melhorar a detecção. O filtro de emissão deverá manter-se em qualquer caso na banda de excitação pois o espectro de emissão não será alterado.

3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

Outra fonte usada no microscópio de fluorescência é a *lâmpada em arco de xénon* que tem um espectro contínuo na zona do visível^[7]. Estas lâmpadas são usadas em sistemas onde as características espectrais das camadas são analisadas mas não são tão brilhantes como a lâmpada de mercúrio com características equivalentes. Mesmo em regiões de excitação onde a lâmpada de mercúrio é fraca, a lâmpada de xénon é mais brilhante. O resultado é a produção da luz na lâmpada de xénon ser aproximadamente o dobro da lâmpada de mercúrio equivalente, o que reduz a quantidade de luz focada na amostra. A comparação das lâmpadas de xénon e mercúrio é apresentada na tabela 3.1^[7].

LAMP TYPE	RATED POWER (watts)	LUMINOUS FLUX (lumens)	AVERAGE BRIGHTNESS (candela/mm ²)	ARC SIZE w x h (mm)	RATED LIFE (hours)
<i>Mercury:</i>					
HBO 50W/3	50	1300	900	0.20 x 1.35	200
HBO 100W/2	100	2200	1700	0.25 x 0.25	200
HBO 200W/2	200	10000	400	0.60 x 2.20	400
<i>Xenon:</i>					
XBO 75W/2	75	950	400	0.25 x 0.50	400
XBO 150W/1	150	3000	150	0.50 x 2.20	1200

Tabela 3.1 – Comparação entre lâmpadas de mercúrio e xénon^[7]

3.2.2. Lasers

Os *lasers*, provavelmente a fonte de luz mais empregue em microscopia, são fontes monocromáticas de grande intensidade, são também usados como ferramenta para uma variedade de técnicas incluindo armadilhas ópticas, estudos de tempo de vida da imagem projectada, recepção de fotografias e fluorescência de reflexão total interna. Os *lasers* são a fonte de luz mais comum para os microscópios *CLSM* e são utilizados com menos frequência em investigações de fluorescência de campo amplo. Os *lasers* emitem feixes intensos de luz monocromática coerente e colimada para formar um feixe firme com uma taxa de expansão baixa. Comparando com outras fontes de luz, os *lasers* emitem radiações com λ puro tendo uma relação largura de banda/fase sem paralelo comparativamente com as lâmpadas de descarga em arco de tungsténio/halogéneo. Como resultado, os feixes de luz *laser*, podem viajar longas distâncias e expandem-se para o preenchimento de aberturas ou para focar um ponto muito pequeno com elevado grau de brilho^[54].

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

A coerência da luz monocromática produzida pelos *lasers* introduz problemas em aplicações com microscópios de campo amplo clássico. A luz dispersa, difracção e as poeiras introduzem interferências em todas as superfícies do caminho óptico. Estes efeitos podem ser minimizados por diversas técnicas. O método mais comum usa luz laser giratória temporária por variação rápida do comprimento dos caminhos ópticos entre a fonte de luz e o microscópio. Outro método usa o varrimento da amostra ponto a ponto como é o caso dos sistemas *CLSM*^[54]. As interferências podem ser eliminadas pela abertura dos diafragmas e por técnicas de varrimento.

A estabilidade de um laser é um aspecto importante em muitas aplicações. Há vários factores que estimulam o ruído do feixe como as flutuações no comprimento da cavidade, mas há outras perturbações que causam flutuações na amplitude criando grande intensidade de ruído ou pequenas variações (*drift*) de potência na saída. Algumas destas flutuações na intensidade são devidas à cabeça laser ou à fonte de alimentação^[29,54].

3.2.2.1 Tipos de lasers

Os *lasers* podem ser divididos nas seguintes famílias:

- *Laser* gasoso;
- *Laser* de estado sólido;
- *Laser* de *dye*;
- Díodo *Laser* (semicondutor).

De entre os *lasers* gasosos refira-se os *lasers* de ião árgon (*Ar*) e kripton (*Kr*) que produzem linhas de emissão múltiplas a partir de transições do gás que diferem em níveis de energia e só poucas linhas são válidas para aplicações de microscopia^[54]. Os *lasers* de ar frio de ião *Ar* são empregues como fontes de luz para o microscópio confocal devido ao seu brilho, ao pequeno tamanho, à excelente geometria do feixe e à estabilidade das linhas do espectro de excitação para fluorescência. Muitos dos *lasers* de ião *Ar* utilizados no microscópio confocal de fluorescência emitem só duas linhas que representam 75% da potência total do *laser*. *Lasers* de ião *Ar* equipados com espelhos podem emitir λ no ultravioleta e no visível^[54]. O *laser* de ião *Kr* tem menos

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

aplicações do que o *laser* de *Ar* devido ao seu elevado λ . Quando um sistema com ião *Kr* produz a potência máxima dá-se um grande aquecimento necessitando então de ser arrefecido artificialmente para gerar a mesma potência do que o *Ar*. A esperança de vida dos *lasers* de ião é reduzida devido ao consumo de gás. O gás é infiltrado nas paredes dos tubos de descarga como resultado das densidades de correntes elevadas necessárias para as operações *laser*^[29,54].

Os *lasers* arrefecidos usam misturas de *Ar/Kr* e tornaram-se populares nos microscópios confocais usando mais do que um λ de iluminação em que são necessários dois ou mais fluoróforos para o seu estudo. Tais *lasers* de misturas de gases só são capazes de produzir saídas estáveis em linhas bem separadas no espectro de λ . Todos os *lasers* do ião mostram uma boa qualidade de feixe e podem ser comprados em diversos fabricantes com uma configuração em linha ou múltiplas linhas^[54].

Os *lasers* de hélio (*He*) e cádmio (*Cd*) são considerados membros da família *He/Ne(neón)* e são uma alternativa económica nas regiões espectrais contínuas no violeta e ultravioleta. Estes *lasers* confiam no vapor de *Cd* como meio *lasing* que se distribui uniformemente durante em todo o orifício desde o reservatório de *Cd*, aquecido pela electroforese em estado gasoso. A bomba de *He* mantém a pressão constante superior ao vapor de *Cd*. Os *lasers Cd/He* têm maior ruído do que o *He/Ne*, porque as flutuações localizadas no vapor de *Cd* são variações de concentração^[54].

O *laser* de nitrogénio (*N*) foi usado como fonte de luz em impulsos para espectroscopia e microscopia. A saída é única com duração de impulsos que varia desde a ordem do picosegundo (*ps*) até ao nanossegundo (*ns*). A taxa de repetição dos impulsos pode ser no máximo de 200*impulsos/segundo*^[29,54]. Os *lasers* de *N* são empregues para bombear moléculas da camada de emissão de grandes λ . Além disso, estes *lasers* foram utilizados como fonte de luz para obter imagens projectadas a grande velocidade, usando uma camada de excitação no ultravioleta. Nesta aplicação, são usados dois *lasers*: um funciona como fonte directa de luz e o outro produz emissão. Pulsando cada *laser* a uma taxa de 15 *repetições/segundo* em sincronia com a taxa de vídeo o mesmo produz uma imagem.

O díodo *laser* está disponível com potências de saída adequadas para a área de microscopia. O mais comum deles opera perto do infravermelho, do vermelho e azul

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

tendo sido desenvolvidos em outros espectros. Os díodos *laser* exibem feixes mais estáveis, permitindo-lhes que substituam os *lasers* de *He/Ne* em muitas aplicações^[54]. Têm uma esperança de vida elevada mas são extremamente sensíveis ao choque electrostático, logo devem ser manuseados com cuidado. O interesse na área da microscopia é o desenvolvimento dos díodos *laser* ajustáveis que podem competir em potência e versatilidade com o *laser* de camadas ajustáveis como os *lasers* de *Safira* e titânio (*Ti*). Têm a vantagem do baixo custo, compactos, elevado tempo de vida e baixa produção de calor.

Os *lasers* de estado sólido de bomba de díodo (*DPSS*) utilizam o díodo *laser* em vez de material *lasing*^[54]. A potência de saída, a qualidade do feixe e a estabilidade exibida pelos *DPSS* aproxima-se dos *lasers* gasosos (*He/Ne*), mas a eficiência e o tamanho são mais parecidos com os díodos *laser*. As operações típicas e os custos de manutenção do *DPSS* são inferiores aos *lasers* de gás e muitos dos sistemas são arrefecidos por ar^[54]. O desenvolvimento do *DPSS* foi direccionado para aplicações industriais e comerciais que requerem grande potência em escalas de λ no verde e ultravioleta. A saída na região do ultravioleta é pulsada, com energias variáveis, com impulsos da ordem do *ns* e taxa de repetição tão elevada (*10kHz*)^[54]. Estes *lasers* são utilizados em microscopia para provocar a libertação de compostos presos. As taxas de repetição de impulsos são demasiado lentas para se utilizar como uma fonte de iluminação para muitas das aplicações do microscópio confocal. O maior desenvolvimento conduziu à combinação de *DPSS* com reguladores ópticos paramétricos produzindo na saída um impulso ajustável^[54]. Embora o sistema inicial seja dispendioso e complexo de operar, foram introduzidas versões de escalas decrescentes mais apropriadas para o uso em microscópios.

O *laser* de safira dopado de titânio (*Ti:Sa*) fornece a vantagem de *tunability* para impulsos que fornecem luz contínua com a fiabilidade do estado sólido. Estes *lasers* fornecem impulsos de luz de pequena duração a taxas de repetição elevadas. Os intervalos de λ ajustáveis estendem-se desde o vermelho ao infravermelho^[29,54]. Muitos destes *lasers* operam com bombas ópticas¹ por *lasers* de *Ar* de grande potência que

¹ São dispositivos compostos por cristais, sintéticos ou não, cujas faces são paralelas entre si, perpendiculares ao eixo e polidas de modo diferente para constituir uma cavidade ressonante. Os cristais são instalados no eixo de uma lâmpada de descarga gasosa helicoidal que funciona em regime pulsado e é responsável pelo bombeamento óptico.

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

necessitam de refrigeração. Como resultado da despesa envolvida com a operação e manutenção de laser *Ti:Sa* o seu uso foi limitado inicialmente ao microscópio multifotónico (*Multiphoton Microscopy*). Recentemente um laser de díodo bombeado por *Cr:LiSAF* (crómio dopado com lítio, estrôncio, alumínio de fluoreto) foi desenvolvido para fornecer grandes frequências de impulsos a um dado λ , uma potência média baixa, pequeno tamanho tornando-se numa fonte de luz atractiva para microscópios de fluorescência de multifotões^[54].

3.3. Tipos de detectores de luz visível

Na fluorescência moderna e no *CLSM* a colecta e a medida da emissão secundária recolhida pela objectiva pode ser realizada por várias classes de detectores fotossensíveis:

- Tubos fotomultiplicadores (*PMT*);
- Fotodíodo;
- Fotodíodo de avalanche;
- Dispositivos de estado sólido de cargas acopladas (*CCD - charge coupled device*);
- Dispositivos de semiconductor de óxido de metal complementar (*CMOS – complementary metal oxide semiconductors*)^[30].

No *CLSM*, a emissão de fluorescência é direccionada através da abertura do *pinhole* posicionado perto do plano da imagem para excluir a luz vinda da estrutura fluorescente localizada longe do plano focal da objectiva, reduzindo a quantidade de luz disponível para a formação da imagem. Como resultado, os níveis de luz muito baixos encontrados num microscópio confocal obrigam à utilização de detectores de fótons com sensibilidade elevada que não necessitam de discriminação espacial mas sim de uma resposta rápida com um grande nível de sensibilidade ao fluxo contínuo de intensidades de luz variável.

3.3.1. Tubos fotomultiplicadores (*PMT*)

O *PMT* usa o efeito fotoelétrico para converter fótons em elétrons. Contém uma superfície fotosensível que capta fótons incidentes e produz uma corrente de fotoelétrons que gera uma carga elétrica amplificada. A cascata de elétrons secundários emitidos amplificam estes fotoelétrons. É o detector mais popular nos *MC* e têm o elemento crítico chamado fotocátodo capaz de emitir elétrons através do efeito fotoelétrico quando exposto a um fluxo de fótons. A anatomia geral de um *PMT* (figura 3.1) consiste num tubo de vácuo de vidro, ou quartzo, que encerra o fotocátodo e a corrente de multiplicação dos elétrons seguido de um ânodo para completar o circuito elétrico. Quando o *PMT* está a funcionar, a corrente que flui entre o ânodo e o potencial zero é proporcional ao fluxo de fotoelétrons gerado pelo fotocátodo quando exposto à radiação de fótons incidentes. Os fotocátodos são fabricados com metais alcalinos e projectados para responder numa região de λ ditado pela composição do material base utilizado na sua construção. A sensibilidade ao λ de iluminação incidente é referida como resposta espectral (R) do fotocátodo:

$$R = (q \cdot \Phi) / h \cdot c \quad (\text{eq.3.1})$$

q – Carga do electrão;

Φ – Eficiência quântica da superfície fotossensível;

h – Constante de *Planck*;

c – Velocidade da luz.

A eficiência quântica do detector é uma medida da fracção de fótons incidentes que resulta numa saída detectável dado pela equação 3.2. A composição do fotocátodo determina não só a resposta espectral do *PMT* mas também a eficiência de cada λ , a total uniformidade da sensibilidade do *PMT* e a existência de corrente negra. Existe uma diversidade de fotocátodos com diferentes eficiências quânticas e gamas de λ . Esses valores estão tabelados para uma melhor escolha do fotocátodo que melhor se coaduna às necessidades de cada aplicação. Os dínodos geram um ganho de ruído que excede o valor de 10^6 com largura de banda elevada^[41,47]. Devido a este ganho, o *PMT* pode operar a níveis baixos (baixo número de fótons detectados por segundo) ou a níveis altos (biliões de fótons detectados por segundo). São particularmente úteis para

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

medidas que são feitas num certo intervalo de tempo, a grande frequência e para sinais ópticos fracos em que os amplificadores externos não podem ser usados ou é necessário uma gama dinâmica muito grande (10^5). O fotocátodo determina a resposta espectral do *PMT* e há fotocátodos comerciais que estão disponíveis com vários λ . A gama de eficiência quântica varia entre 10% a 20%. As superfícies fotossensíveis podem ser muito grandes num máximo de 500mm^2 ^[45,47]. É possível que a curva da resposta espectral de um *PMT* espacial possa variar devido a variações individuais das composições do fotocátodo.

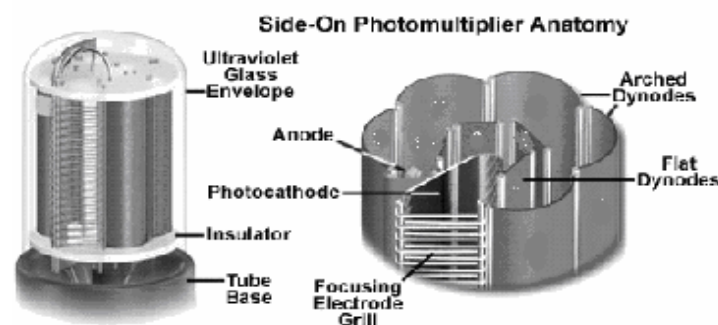


Figura 3.1 – Representação geral de um fotomultiplicador^[54]

3.3.2. Fotodíodo

O fotodíodo é um componente semiconductor de estado sólido. Um fóton com energia maior do que a largura de banda de um semiconductor excita um electrão para a banda de condução e cria uma lacuna na banda de valência. Uma tensão aplicada desloca o par electrão/lacuna em direcções opostas gerando uma corrente eléctrica. O fotodíodo não tem um ganho sendo o seu sinal tão pequeno que é necessário um amplificador externo^[30]. Devido à introdução de ruído pelos amplificadores, os fotodíodos são usados em medidas de níveis de luz elevadas onde os detectores de fótons têm maior ruído do que o dos amplificadores externos logo alcançam o maior valor de *SNR* quando operam em níveis de luz elevados. Os fotodíodos também são usados para frequências menores do que os *PMT* para reduzirem o ruído amplificado. Uma vantagem dos fotodíodos é que operam a níveis de luz muitas vezes maior do que os permitidos pelos *PMT*. Eles são robustos e menos dispendiosos. Os fotodíodos são

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional válidos para gamas de espectros variáveis dependendo do material do semiconductor^[47]. O pico de sensibilidade do fotodíodo é conseguido a $900nm$ e a resposta medida em intensidade de corrente por unidade de potência de radiação incidente é cerca de $0.5A/W^{[47]}$. A sensibilidade dos fótons do díodo é definida através da eficiência quântica $\Phi^{[47]}$.

$$= 100 \frac{\text{fotoelectrão}}{\text{fótons incidentes}} (\%) \quad (3.2)$$

A $500nm$ tem-se 2.25×10^{18} fótons por joule de energia, mas a $800nm$ tem-se 4.03×10^{18} fótons por *joule*^[47]. Pode-se ver um fotodíodo na figura 3.2.

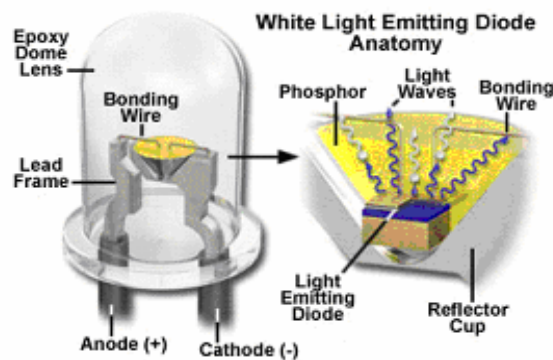


Figura 3.2 – Anatomia de um fotodíodo^[54]

3.3.3. Fotodíodo de avalanche

O fotodíodo de avalanche (*APD*) é um detector que combina a grande eficiência quântica com o elevado ganho do *PMT*. O *APD* é um semiconductor de estado sólido que gera um par electrão/lacuna depois de exposto à luz. Contudo, em contraste com um fotodíodo convencional, uma polarização reversa é aplicada a uma junção *pn* criando um campo eléctrico interno suficientemente grande para a aceleração de electrões^[41,47]. Quando colidem com os átomos de valência os electrões acelerados geram um novo par electrão/lacuna conduzindo à criação de uma avalanche de electrões emitindo assim um sinal amplificado com ganhos elevados^[41]. As flutuações estatísticas no número de colisões e o rendimento de pares electrão/lacuna criados vai criar mais ruído do que os *PMT* mas menos ruído do que um amplificador fotodíodo. O ganho resultante é menor do que o do *PMT* desde que a eficiência quântica do *APD* seja

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

aproximadamente uma ordem de grandeza maior (p.ex. um *APD* com um ganho de 10^3 corresponde a um *PMT* com um ganho de 10^4). O *APD* sofre de correntes negras elevadas e o seu ganho depende da temperatura de operação que tem de ser mantida fixa^[41]. O modo de detecção mais sensível de um *APD* é o modo de contagem de fótons simples. Desta forma não é medida a corrente do fotodetector, mas os impulsos de corrente causados pela incidência de fótons. A discriminação do ruído é feita por ajuste do ponto inicial de corrente, em que apenas os impulsos que passam esse valor de corrente são contabilizados. Hoje os modelos de díodos de avalanche de fótons simples (*SPDA* – *single photon avalanche diode*) têm eficiência quântica maior do que os *PMT*, o que os torna competitivos com os parcialmente superiores *PMT*^[30,45]. A peculiaridade dos modelos *SPDA* é que todos os fótons detectados geram pequenas quantidades de luz de luminescência dentro do cristal de diodo com uma larga distribuição do espectro. Isto pode causar dificuldades quando empregamos vários *SPDA*, ou se a luminescência da luz reflectida gerar um sinal no próprio *SPDA*. Se os fótons de luminescência são detectados após o tempo morto do *SPDA*, podem gerar eventos de detecção falsos^[41,47]. Podem-se ver as diferenças entre um fotodíodo de avalanche e um *PMT* na figura 3.3.

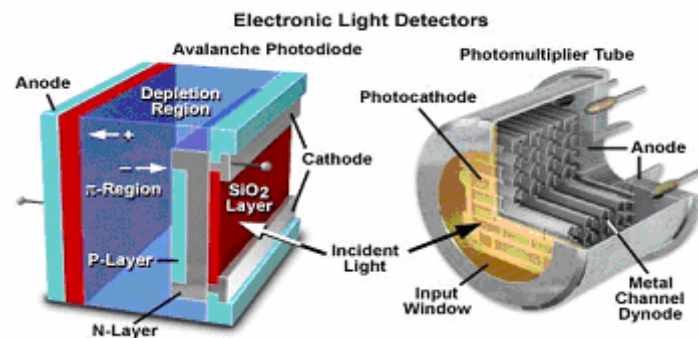


Figura 3.3 – Comparação de um fotodíodo de avalanche e o fotomultiplicador^[54]

3.3.4. Sensor CCD

Os sensores de imagem *CCD* são capazes de transformar um modelo de luz num modelo de carga eléctrica. O *CCD* é um conjunto de elementos que têm a potencialidade de colectar, armazenar e transportar a carga eléctrica de um elemento para outro^[49]. Cada elemento fotosensível representará um elemento da imagem (*pixel*).

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

Com tecnologias de semicondutor e regras rígidas de projecto, as estruturas são feitas em forma de linha ou matrizes de *pixels*. Os amplificadores colectam à saída do *chip* os sinais do *CCD*. Uma imagem electrónica pode ser obtida, depois da exposição do sensor a uma imagem, pela aplicação de impulsos que transferem a carga de um *pixel* após outro para o amplificador de saída que converte a carga em tensão^[49]. A electrónica transforma o sinal de saída num modelo apropriado para ser monitorizado. Os *CCD*'s obtêm figuras com ruído baixo^[49].

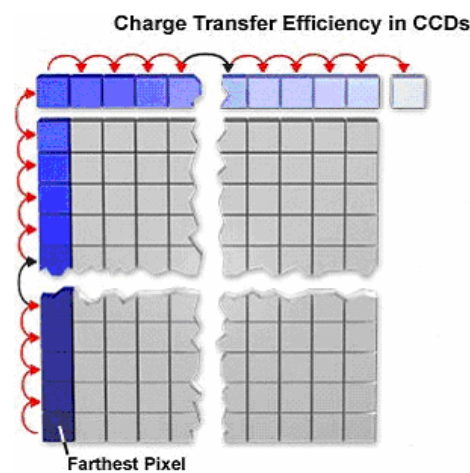


Figura 3.4 – Exemplo da transferência de cargas num *CCD* de *array* linear e em área^[54]

Os *CCD*'s foram muito usados em medidas sem contacto e processamento de imagens^[49]. Podem ser classificados como de matriz (*array*) linear ou em área. Pode-se ver na figura 3.4 transferências de cargas numa matriz linear e em área. O *CCD* de matriz em área pode ser usado directamente em medidas de posições a duas dimensões e tem menos pixels em cada direcção do que um *CCD* de matriz linear e o circuito de processamento de sinal é mais complexo tendo um maior tempo de resposta^[48]. Um *CCD* de matriz linear é normalmente usado apenas para medidas de posição a uma dimensão e tem as seguintes características^[48,50]:

1 – *Resposta rápida* – O período de processamento de sinal é pequeno o que o torna mais apropriado para medidas de ponto móvel;

2 – *Elevada precisão* – A precisão das medidas é influenciada pelo número de *pixels* do *CCD* e pode-se obter grande precisão no sistema quando os componentes ópticos são alinhados apropriadamente;

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

3 – *Baixo custo* – Um *CCD* de matriz linear tem mais *pixels* numa direcção e um preço mais baixo do que um de matriz em área. Assim fornece uma opção em termos de performance/custo mais atractiva para medidas de posição a duas dimensões.

Os sensores *CCD* podem ser de cor ou monocromáticos^[49]. Nos de cor o filtro é integrado e fornece a resposta em cor e no monocromático é detectado a preto e branco. As escolhas para o tipo da disposição incluem a disposição linear, disposição da área de transferência da *frame*, disposição da área total da *frame* e disposição da área de transferência da linha interna^[49]. O formato óptico é a medida do tamanho da área total da imagem reflectida. O formato óptico é usado para determinar o tamanho da lente necessária para a utilização do filtro (ou *toner*) a integrar no *CCD* e refere-se ao comprimento na diagonal da área da imagem projectada. As escolhas ópticas do formato incluem distâncias de 0.363cm a 2.540cm^[50]. É importante definir o número de *pixels* e o seu tamanho. Os *pixels* horizontais referem-se ao número de *pixels* na linha e os *pixels* verticais referem-se ao número de *pixels* na coluna do sensor. Quanto maior é o número de *pixels* melhor a definição do *CCD*. A definição de *VGA* é 640x480 significa que o número de *pixels* horizontais é 640 e verticais é 480^[50]. Pode-se ver um *CCD* na figura 3.5.

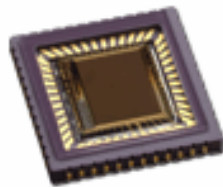


Figura 3.5 – Exemplo de um *CCD*

É importante conhecer as especificações e desempenho de um *CCD* quando da sua escolha. As especificações incluem a resposta espectral, taxa de dados, eficiência quântica, escala dinâmica e o número das saídas^[49]. A resposta espectral é a escala de λ para que é projectado. A taxa de dados é a velocidade de transferência de dados (*MHz*). A eficiência quântica é a relação dos electrões/fotões gerados que o *pixel* capta pelo número de fotões incidentes na área do *pixel* (eq. 3.2). A escala dinâmica é a relação logarítmica da profundidade em função do ruído (*dB*). As características comuns para os *CCD* incluem antifuorescência e refrigeração^[49]. Algumas disposições dos *CCD*

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

oferecem uma porta antifuorescência opcional projectada para eliminar o excesso de luz num *pixel* saturado^[50] que pode causar uma raia vertical^[50]. Algumas disposições de *pixels* são refrigeradas para se obterem valores de ruído mais baixo e sensibilidade mais elevada.

3.3.5. Sensor CMOS

Os sensores *CMOS* operam a tensões inferiores ás do *CCD*, reduzindo o consumo e a potência^[51]. Cada célula activa do *pixel* do *CMOS* tem o seu próprio amplificador, podendo ser lido individualmente^[51]. Uma célula geralmente tem quatro transístores e um elemento fotossensível. A célula tem uma porta de transferência que separa o fotossensor de difusão flutuante, uma porta de *reset* entre a difusão flutuante e a fonte de alimentação, um transístor de fonte seguidor para proteger a difusão flutuante da capacidade de leitura da linha e uma porta para seleccionar a coluna para conectar a célula à linha da leitura. Todos os *pixels* numa coluna estão conectados a um amplificador^[51]. Além do consumo e da potência ser mais baixo comparado com o *CCD*, o *CMOS* é muito mais simples de projectar requerendo menos circuitos^[52].

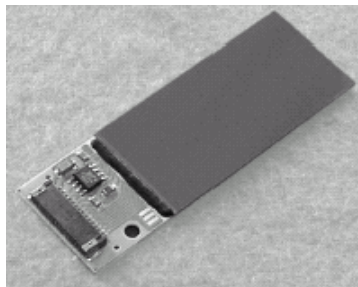


Figura 3.6 – Exemplo de um *CCD*

Há duas categorias de sensores *CMOS*, analógicos e digitais^[51]. Os analógicos conectam o seu sinal codificado num formato vídeo, podendo o sinal alimentar directamente o equipamento de vídeo. Os *CMOS* digitais fornecem um sinal digital através de um bus de 4 a 16 *bits* directo e não necessita da conversão através de um processo de descodificação^[51]. O vídeo, o equipamento da imagem e os componentes

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

de varrimento são todos dispositivos usados na tecnologia vídeo para gravar ou ler dados através de acções realizadas na imagem reflectida^[51].

Em aplicações industriais são utilizados para a inspecção, medidas, controle de qualidade, detecção da imagem, leitura, análise e indicação de dados. O vídeo, o equipamento da imagem projectada, os componentes de varrimento são divididos em sete famílias diferentes: equipamento da visão, câmaras e lentes, equipamento vídeo, sensores da imagem, equipamento de código de barra, medidores, leitores e indicadores^[52]. Tem-se o exemplo de um *CMOS* na figura 3.6. O equipamento vídeo inclui dispositivos para processar dados. Esta família inclui registradores de vídeo digital utilizados para transformar os dados vídeo na saída digital, monitores, multiplexadores para lidar com um certo número de sinais de uma vez e comutadores para controlar a sequência da câmara vídeo^[52].

3.3.6. CCD versus CMOS

Os sensores de imagem *CCD* e *CMOS* são duas tecnologias diferentes para captar imagens. Cada um tem características típicas que lhe dão vantagens em relação ao outro em determinadas aplicações. Nenhum é superior ao outro, embora os vendedores que vendem somente uma tecnologia reivindicem o contrário. Num *CCD*, a carga de cada *pixel* é transferida para um número muito limitado de nós de saída para ser convertido num sinal analógico e emitido em tensão^[53]. Todo o *pixel* é direccionado para a captação de luz, tendo a saída grande uniformidade, um factor chave na qualidade da imagem. Num *CMOS*, cada *pixel* tem sua própria conversão da carga em tensão que inclui amplificadores, correcção de ruído e circuitos digitais para que a saída do *chip* seja digital^[49,51]. Estas funções aumentam a complexidade do projecto e reduzem a área disponível para a captação de luz. O *chip* pode ser construído para diminuir o número de circuitos fora dele.

Nos últimos anos verificaram-se alterações significativas em ambas as tecnologias. Actualmente existe uma estrutura nova, os *toners* de *CCD* e de *CMOS*, que apresentam novas oportunidades e características especiais^[53], que foram inventados na década de 60^[49,51]. Ambos os tipos de *toners* convertem a luz em carga eléctrica e

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional processam-na em sinais electrónicos^[53]. O *CCD* tornou-se dominante porque fornecia imagens superiores com a tecnologia disponível na altura. Os sensores *CMOS* requeriam mais pormenor a uma menor escala do que as fundições de *chips* de silício poderiam fornecer. Na década de 90, a litografia desenvolveu-se ao ponto que os desenhadores pudessem dar mais importância aos *toners* dos *CMOS*^[51]. O interesse renovado no *CMOS* foi baseado em expectativas de consumo a baixa potência, integração da câmara num *chip* e produção a baixo custo^[52]. Concretizar todos estes benefícios na prática, fornecendo imagens de qualidade elevada e menos dispendiosas, foi mais fácil do que inicialmente se pensou. Os *toners* de *CCD* e de *CMOS* podem oferecer um excelente desempenho na imagem quando projectados correctamente. Os *CCD's* fornecem um desempenho superior ao nível fotográfico, científico e em aplicações que exigem maior qualidade de imagem medida pela eficiência quântica mas à custa do aumento do sistema^[49]. Os *toners* de *CMOS* oferecem mais integração, ou seja, mais funções num único *chip* diminuindo a dissipação de potência mas requer compensações entre a qualidade da imagem e o custo do dispositivo^[53]. Os projectistas de *CMOS* direccionaram o esforço para conseguir elevada qualidade na imagem enquanto os de *CCD* baixaram as exigências de potência e o tamanho do *pixel*. Podem-se encontrar *CCD's* nas câmaras de baixo consumo e custo, em telemóveis e sensores *CMOS* em câmaras de grande performance e industriais que contradizem as aplicações iniciais. Os custos ao nível do *chip* entre eles são similares. Inicialmente propõe-se que os *toners* dos *CMOS* seriam mais baratos porque poderiam ser produzidos nas mesmas linhas de processamento do *chip*, em grandes quantidades como as memórias, mas não foi o caso. As condições necessárias para a boa performance da imagem projectada requerem *CMOS* especializados, com optimização do processo de fabrico como os usados nos *CCD's*^[49], provocando, em litografia, nós sucessivamente menores fora deste processo o que os torna lentos e caros^[49]. As câmaras *CMOS* requerem poucos componentes e menor potência mas ainda necessitam de *chips* auxiliares para optimizar a qualidade de imagem, aumentando o custo e reduzindo as vantagens.

Os *CCD's* são menos complexos e custam menos a projectar e fabricar, estando esses processos mais maduros e optimizados o que torna o *CCD* mais barato^[53]. O tamanho do *chip* pode ser um factor dominante no custo do dispositivo pois quanto

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

maior o *chip* mais dispositivos pode conter. A produção em grande escala de *CMOS* para certas aplicações fixou um preço baixo. A velocidade é uma área onde os *toners* de *CMOS* podem demonstrar uma força considerável devido à facilidade de estruturas paralelas na saída.

3.4. Técnicas de microscopia tridimensional

Em microscopia tridimensional podem ser aplicadas várias técnicas de entre as quais seleccionamos e vamos descrever as seguintes:

- Microscópio confocal de fluorescência
- *PAM (Programmable Array Microscopy)*
- Microscópio Confocal de Linha (*CL*)
- Microscópio de Campo Amplo com iluminação estruturada;
- Microscópio de varrimento *Tandem* (com disco de *Nipkow*);

Cada técnica tem um desempenho em que sobressaem algumas características que são vantajosas para determinadas aplicações específicas limitando a sua utilização em outras aplicações.

3.4.1 Microscopia confocal de fluorescência

Já se descreveram os seus aspectos mais importantes no capítulo anterior faltando agora referir que existem duas principais variações dos mecanismos de varrimento usualmente empregues para produzir as imagens do microscópio confocal:

- Varrimento da plataforma da amostra juntamente com um feixe de luz de iluminação estacionária (varrimento do objecto);
- Varrimento de um feixe de luz com a plataforma da amostra estacionária (varrimento do feixe);

A sua grande aplicação deverá ser como microscópio confocal de fluorescência em que são utilizados para se obter imagens de espécimes^[2] em que previamente se usou um marcador de fluorescência adequado. Em contraste com o

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

microscópio de fluorescência convencional, as configurações confocais permitem o seccionamento óptico do objecto, através da iluminação com uma fonte pontual, sendo a detecção feita por um *pinhole* na posição conjugada no percurso de emissão obtendo-se imagens a três dimensões^[14,36]. Para adquirir a imagem total é necessário que haja o varrimento da fonte de luz em toda a amostra. O ciclo para varrer toda a amostra no *CLSM* é grande uma vez que o laser só ilumina um ponto de cada vez.

3.4.2. PAM

O *PAM* usa um modelador espacial de luz (*SLM*) para gerar um modelo de iluminação conjugada. O *SLM* divide a luz de fluorescência refractada pela objectiva numa imagem focal conjugada (I_c), formada pela luz no foco, e a imagem não conjugada (I_{nc}), formada pela luz fora do foco. Existem duas diferentes arquitecturas para a utilização do *PAM*^[36]. A primeira tem uma grelha de pontos que se desloca para varrer a imagem na sua totalidade. A segunda, adiante designada por sequência, consiste num pequeno varrimento da amostra em dois eixos paralelos formando uma imagem a duas dimensões. No primeiro caso I_c é análoga à imagem confocal, e I_{nc} é análoga a uma imagem convencional menos I_c . No segundo caso I_c e I_{nc} são o somatório e a diferença da imagem convencional e da imagem confocal. A aproximação da sequência necessita de um processamento posterior para recuperar a parte confocal, indo gerar um sinal de nível elevado para um tempo de integração também elevado.

Os *SLM* são classificados de acordo com o modo de operação em transmissão ou reflexão. Um exemplo é o dispositivo de microespelhos digital (*DMD*) que consiste numa matriz de espelhos quadrados em que cada um pode ser inclinado em relação à normal^[36,38]. A figura 3.7 representa um esquema de um microscópio de fluorescência com um *array* programável de *DMD*. O sistema é iluminado uniformemente por uma fonte de luz. Os espelhos colocados na posição 'on' reflectem a luz do objecto na posição conjugada do plano focal pretendido. A emissão de fluorescência é verificada atrás dos elementos correspondentes do *DMD* e reflectida, por via óptica, para um detector (câmara 1), correspondendo à imagem conjugada focal. A fluorescência

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

proveniente de fora do foco, originada a partir de planos não focais é reflectida pelo espelho na posição ‘off’ e observada no segundo detector (câmara 2).

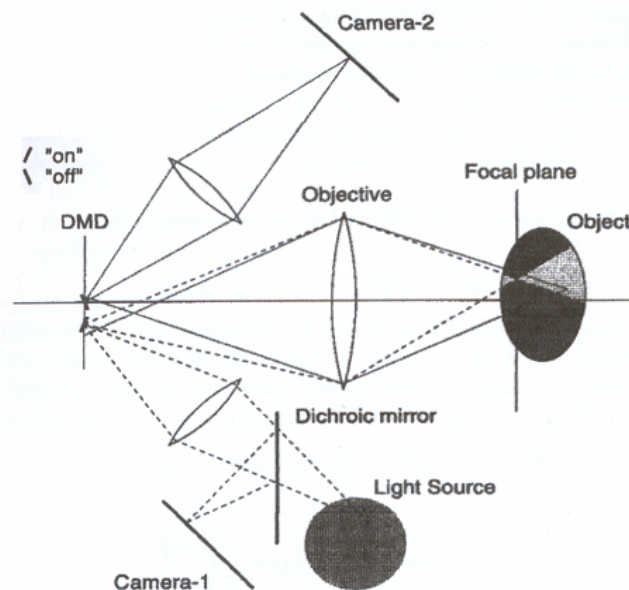


Figura 3.7 – Esquemático de um microscópio *PAM* de fluorescência baseado num *DMD*^[36]

Podem-se retirar as seguintes conclusões acerca das duas estratégias para formação da imagem confocal com o *PAM*:

1) Em comparação com o *CLSM*, o *PAM* aumenta a velocidade de aquisição em duas ordens de grandeza com o aumento do sinal em uma ordem de grandeza. Apesar de a lâmpada da fonte de luz do *PAM* produzir uma menor irradiação no plano focal do que o laser no *CLSM* isto é compensado pelo grande número de pontos que são varridos em paralelo. O *PAM* tem grande eficiência quântica de detecção (sensor *CCD* em vez do *PMT*).

2) A aproximação da sequência pseudoaleatória é de grande eficiência luminosa, que resulta do facto de 50% dos elementos *SLM* estão ‘on’ em cada modelo de modulação. Por outro lado ambas aproximações produzem a mesma imagem confocal para a mesma grelha. A sequência pseudoaleatória gera um sinal muito intenso que é independente do comprimento da sequência. Para obtermos a imagem confocal pela aproximação da sequência pseudoaleatória deve ser subtraído um termo de compensação^[14,36].

3.4.3. Microscópio confocal de linha

A técnica do *CL* combina algumas das características do *CLSM* com métodos de obtenção de imagem convencionais. Baseia-se na iluminação e detecção em forma de linha como alternativa à aproximação do ponto focal corrente (*CP*). O microscópio confocal a uma dimensão tem performances adequadas a várias aplicações sendo adaptável ao crescente número de observações microscópicas. A ausência de componentes móveis no caminho óptico e o uso de imagens electrónicas permite a flexibilidade da velocidade nas operações o que é relevante em muitos campos de investigação. O *CL* é baseado na iluminação em forma de linha e detecção da imagem numa posição conjugada^[40]. O lugar do ponto de iluminação coincide com o ponto focal do microscópio, como se verifica na figura 3.8.

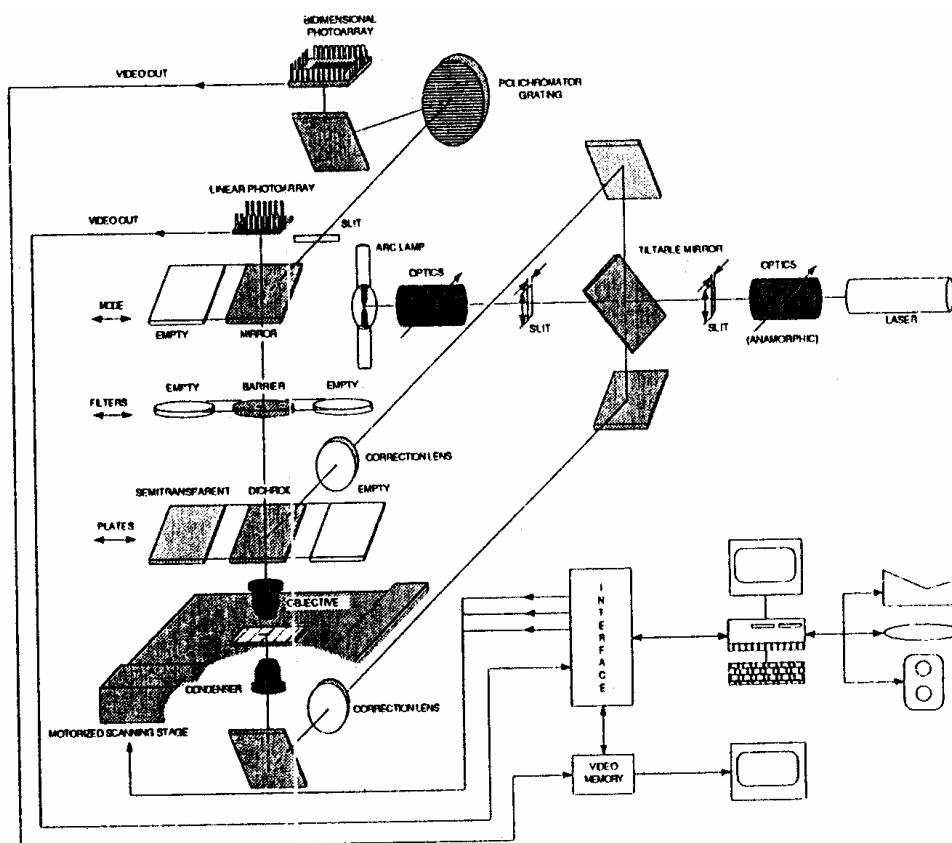


Figura 3.8 – Diagrama óptico do microscópio *CL*^[40].

No *CL* foram melhorados muitos aspectos práticos. O uso de um detector electrónico de imagem, que elimina a necessidade de varrimento mecânico num dos

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

eixos, permite uma análise da área de luz transmitida com associação de técnicas de translação lentas da amostra. Isto é possível pelo uso de níveis de varrimento motorizados. Os campos largos da amostra podem ser explorados e uma grande área pode ser investigada, usando os campos adjacentes de forma a construir uma imagem estendida. Não há raios ou partes ópticas a moverem-se, como as objectivas que são usadas na parte central do seu campo coberto por uma abertura que corresponde à linha. Os tamanhos das linhas e aberturas podem ser facilmente geridas nos dois eixos de forma a obter o melhor compromisso entre profundidade focal e contraste. As fontes convencionais podem ser usadas em muitos casos, devido à diminuição dos requisitos no brilho da fonte. O método do *CL* adapta-se a certas formas do microscópio, como por exemplo a transmissão, reflexão e análise de fluorescência que podem ser realizadas em simultâneo, no mesmo instrumento, se a fonte de radiação for adequada.

O arranjo óptico proposto não é confocal na direcção paralela à imagem mas pode-se obter uma melhoria no contraste. De facto, em comparação com o microscópio convencional em que se obtém a imagem de todo o campo aqui é uma linha que é simultaneamente iluminada e detectada. O método do *CL* produz contrastes que melhores do que o microscópio convencional em termos da imagem projectada transmitida numa estrutura de absorção pequena, quando operados com aberturas grandes, enquanto que os instrumentos confocais correntes não permitem o trabalho em luz transmitida. O contraste melhorado é obtido em microscópios de fluorescência. Acerca da capacidade de seccionamento das técnicas de *CL*, as observações preliminares mostram que a deslocação segue leis que são o meio caminho entre o microscópio *CP* e a imagem reflectida convencional. Mais precisamente, a discriminação em profundidade focal depende do *setup* óptico particular (transmissão, reflexão, etc.) e da interacção entre os dois eixos^[41].

Um exemplo de uma aplicação é a observação de detalhes ópticos das extensões de cromossomas de metafase que em termos da resolução é superior ao microscópio óptico. Alguma melhoria na interpretação do contraste e na qualidade das imagens representam uma vantagem nas análises biomédicas que necessitam de aquisição e armazenamento de uma grande quantidade de figuras.

3.4.4. Microscópio de campo amplo com seccionamento óptico

O microscópio de campo amplo é o microscópio de fluorescência mais comum. A amostra é iluminada totalmente de uma única vez com luz excitada. Como resultado da luz de fluorescência fora do plano focal, tal como a luz de fluorescência das reacções em cima e em baixo da câmara é a luz tremula fora do foco. Mesmo a luz originada a partir de um ponto no plano focal não é exactamente reflectida num ponto devido à difracção da luz pela objectiva. A imagem de um único ponto é função de propagação do ponto e determina a resolução do sistema.

A transformada de *Fourier* da *PSF* de um microscópio de campo amplo mostra falhas nas frequências espaciais ao longo dos eixos ópticos. Por isso os objectos com frequências nessa região irão gerar uma imagem fraca^[43]. Logo a posição da amostra fina de fluorescência em relação ao plano focal não pode ser determinada. Estas falhas são a razão pela qual os microscópios de campo amplo não são indicados para a obtenção de imagens a três dimensões já que não só as frequências elevadas são eliminadas como também umas parte de frequências baixas na direcção axial^[42, 43].

O seccionamento óptico no microscópio de campo amplo pode ser obtido por iluminação do objecto com um movimento contínuo num modelo de frequência espacial única. Este modelo pode ser obtido por projecção de uma grelha na amostra ou utilizando a interferência de duas fontes laser para iluminar a amostra^[14,42,43].

3.4.4.1. Luz estruturada num microscópio convencional

A utilização de luz estruturada é uma técnica recente que resulta num método simples de obter o seccionamento óptico num microscópio de campo amplo. Uma das suas formas é por projecção de uma grelha com frequência espacial única. Apenas partes do objecto, em que a grelha está no foco, são reflectidas no detector. A imagem do seccionamento óptico, com um modelo de iluminação removido, é obtida por meio de um algoritmo de passo de três fases. Consequentemente, três imagens com três diferentes fases retardadas são tidas em conta. Estas imagens retiradas em três posições espaciais da grelha são processadas em tempo real para produzirem imagens de

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

seccionamento óptico similar às obtidas no microscópio confocal^[14,27]. Isto sugere a base de uma aproximação para obter imagens de seccionamento óptico a partir de um microscópio de campo amplo. O microscópio irá determinar a imagem da parte do objecto que se encontra no foco. Obtém-se uma imagem seccionada do objecto com o modelo da grelha imposto. A taxa de atenuação irá depender das frequências espaciais quando projectada no objecto. O ideal é projectar vários modelos dentro do objecto para obter informação acerca de estruturas a 3D com vários métodos^[44,45]. O problema com todas estas aproximações é que deve ser um método simples e eficaz para remover a grelha indesejada a partir da imagem óptica seccionada^[44,45]. A aproximação ao microscópio convencional com o uso de uma dada forma da grelha de detecção, deverá ser alinhada dentro dos requisitos sendo estas configurações facilmente implementáveis. Na fig. 3.9 apresenta-se um esquema de um microscópio com aplicação de luz estruturada em a imagem final é gravada por uma câmara *CCD* no plano da imagem.

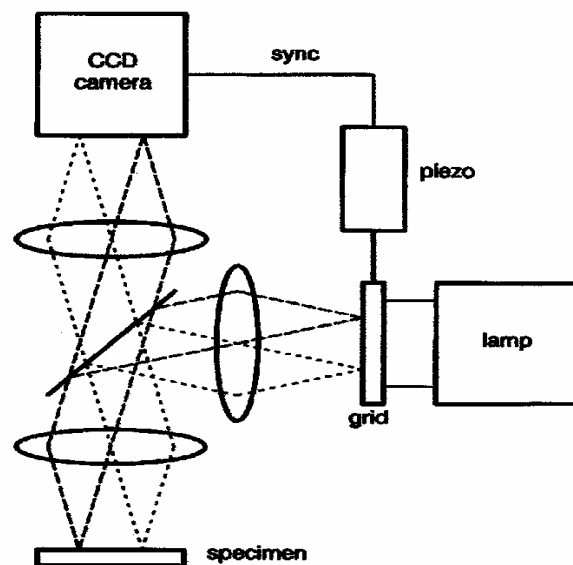


Figura 3.9 – Esquema óptico de um microscópio convencional com luz estruturada^[44].

Conclui-se assim que uma modificação simples de um sistema de iluminação convencional permite obter imagens do seccionamento óptico através de um volume de um objecto fino. A formação da imagem é diferente daquela obtida num microscópio confocal e a função de transferência é determinada para ser diferente para o objecto que varia nas direcções axial e lateral. Contudo, as diferenças não são muito grandes nas

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

amostras que foram examinadas com estas técnicas e obtiveram-se imagens similares às obtidas com um sistema confocal baseado num *laser*.

3.4.5. Microscópio de varrimento *Tandem* (com disco de *Nipkow*)

Trata-se de um novo tipo de microscópio óptico de varrimento confocal em tempo real (*CSOM*) baseado num microscópio óptico de varrimento *Tandem* (*TSOM – Tandem scanning optical microscope*)^[46]. Tem poucos componentes ópticos e é de fácil alinhamento. A sua velocidade e simplicidade, mantendo todas as vantagens de um *CSOM* mas com melhor resolução transversal é aumentado assim o potencial da sua utilização. Estas duas propriedades fazem do *CSOM* uma ferramenta poderosa. Aumentou-se a velocidade do varrimento do microscópio pela utilização do disco de *Nipkow* para varrer muitos pontos perto da amostra como vemos na figura 3.10. O arranjo é similar ao *TSOM* com a excepção que o mesmo conjunto de *pinholes* é usado para iluminar e detectar, simplificando assim o alinhamento. A luz proveniente de uma fonte laser ou de uma lâmpada em arco, ilumina simultaneamente alguns milhares de *pinholes* num disco rotativo que é feito de uma fotomáscara de vidro cromado, contendo *pinholes* com um dado diâmetro e espaçados. Desde que a separação dos orifícios seja suficientemente grande a interacção entre os orifícios vizinhos é negligenciável. Cada *pinhole* ilumina um único ponto do objecto. O movimento de rotação do disco cria um modelo em espiral para preencher completamente o espaço entre os orifícios.

Este microscópio em tempo real tem a mesma resolução transversal que um único *pinhole standard* no *CSOM*. Devido ao espaçamento dos buracos, só 1% da luz iluminada no disco passa através do *pinhole*. Outro problema é eliminar a reflexão de fundo a partir do próprio disco. No sistema usa-se um disco de baixa reflectividade, um polarizador é colocado em frente da lâmpada de arco de mercúrio e um analisador na ocular para eliminar a luz reflectida a partir do disco.

Uma técnica para melhorar a falta de brilho que caracteriza os sistemas que usam o disco *Nipkow* é o uso de microlentes para intensificar a fonte de luz. Como se

_3. Fontes de luz e sensores usados em diversas técnicas de microscopia tridimensional

viu, os sistemas antigos transmitem só cerca de 1% da luz incidente no disco e necessitam de utilizar tanto a refrigeração do *CCD* como câmaras de vídeos para compensar o baixo nível de sinal. As configurações mais recentes do microscópio de varrimento com o disco de Nipkow incorporam um segundo disco contendo milhares de microlentes que rodam em alinhamento com o primeiro disco e amplificam a luz que passa em ambas as direcções através das aberturas desse disco. Alguns modelos mais avançados combinam o disco de rotação duplo e microlentes com iluminação a laser e detectores *CCD* de refrigeração para permitir a aquisição de dados de forma rápida.

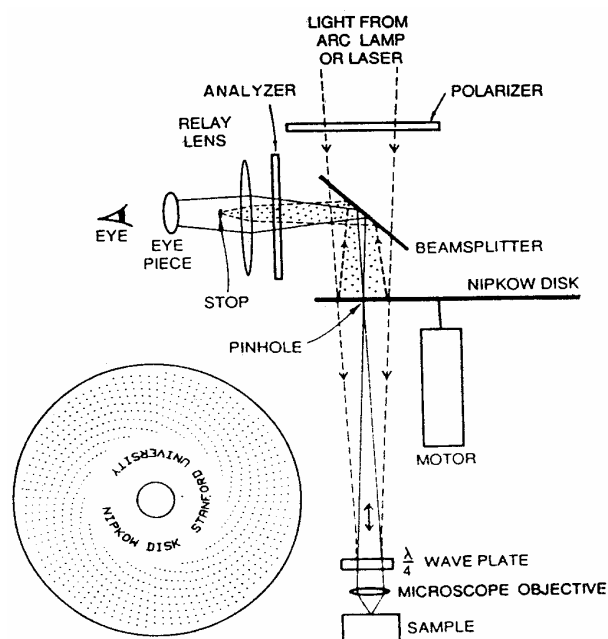


Figura 3.10 – Esquema de um CSOM em tempo real com disco Nipkow)^[46].

4. Módulo de leitura de dados do sensor

4.1. Introdução

Neste trabalho desenvolveu-se um módulo simples e de baixo custo para leitura de dados de um sensor linear do tipo *CMOS* com o objectivo de permitir usar esse sensor para recolher a imagem de um microscópio com varrimento. O projecto da placa de circuito impresso (*PCB*) teve em vista permitir que seja colocada na montagem da mesa óptica, e conseqüentemente a integração do sensor no circuito óptico que consiste numa configuração epi-iluminada de um microscópio com varrimento, que será descrita no capítulo seguinte.

O sensor de imagem do tipo matriz linear de tecnologia *CMOS* é o *LIS-1024* da *Photon Vision Systems* com 1024 *pixels* fotodíodos com uma largura de 7,8 μ m e altura de 125 μ m, em que a frequência de leitura máxima é de 20MHz. A placa desenvolvida é baseada num microcontrolador da *Microchip*, o *PIC16F876* (contém 3 *timers*, um *A/D* de 10 *bits*, *USART*, etc.), que envia os dados lidos do sensor para o *PC* através de um porto série pelo protocolo *RS232*. Desta forma é feita a visualização no *PC* da distribuição da intensidade de luz no sensor em tempo real, gráfico da intensidade em função dos *pixels*, e simultaneamente podem-se guardar os dados para posterior processamento^[57,58]. Este sistema permite obter uma taxa de aquisição de 1 *frame/s* (1 leitura dos 1024 *pixels* por segundo) que é adequada ao objectivo de prova da confocalidade do sistema optoelectrónico, em que ainda não são efectuados os varrimentos que serão necessários quando se pretende construir uma imagem.

A programação do microcontrolador *PIC16F876* foi realizada através das ferramentas de desenvolvimento da *Microchip*. O código foi desenvolvido na linguagem *C* da *Borland* usando-se o compilador *MPLAB* que permite “traduzir” código *C* para *Assembler* que é a linguagem de mais baixo nível para programar o microcontrolador. O *Borland C* também foi utilizado para desenvolver o *software* no

PC que consistiu na interface gráfico, a visualização dos dados do sensor e sua gravação em ficheiros no *PC*.

4.2. Arquitectura do módulo

No projecto e realização do sistema de aquisição dos dados retirados do sistema óptico de bancada, várias hipóteses de arquitectura foram consideradas. Poder-se-ia optar pelo desenvolvimento de um módulo de *PC*, de um módulo autónomo ou adquirir um módulo comercial para um *PC*. A primeira e a última hipótese não vingaram devido ao seu elevado custo e algumas incompatibilidades com outros sistemas tais como sistemas ópticos e os sensores que poderiam ser utilizados. Assim, optou-se pelo módulo autónomo desenvolvido devido às seguintes características^[58]:

→ Rapidez de projecto e realização do sistema principalmente devido ao conhecimento e experiência de todas as ferramentas para o seu desenvolvimento;

→ Baixo custo – a produção deste *PCB* e das suas comunicações ficam, em termos de custo, muito aquém de outras soluções disponíveis no mercado mesmo considerando o custo de desenvolvimento;

→ Possibilidade de integração no sistema óptico – devido à especificidade da aplicação em vista não era possível conseguir esta integração usando os sistemas disponíveis no mercado;

→ Conhecimento do microcontrolador *PIC* da *Microchip* – este último ponto foi decisivo na opção por desenvolver o módulo pois existe um grande conhecimento destes microcontroladores assim como as ferramentas de desenvolvimento de *software* que resultam da sua aplicação noutros projectos deste grupo de investigação.

Como resultado da opção acima apresentada e devidamente justificada, desenvolveu-se um circuito electrónico que culminou numa *PCB* que foi integrado na montagem óptica e que comunica com um *PC* por um porto série, *COM1* ou *COM2*, através do protocolo *RS232*^[57,58] para enviar os dados do sensor, que poderão ser tratados posteriormente. O diagrama de blocos do circuito desenvolvido é mostrado na figura 4.1. O esquema do sistema autónomo foi executado em *TANGO*, podendo ser encontrado na sua forma completa no apêndice A.

A arquitectura da placa autónoma foi desenvolvida tendo como base o objectivo da recolha da informação óptica por parte do sensor LIS-1024 sendo essa informação transferida para um *PC* pelo protocolo *RS232*, usando-se para isso o *driver* *MAX242*. O *PC* controla a transferência de dados pelo *PIC16F876*.

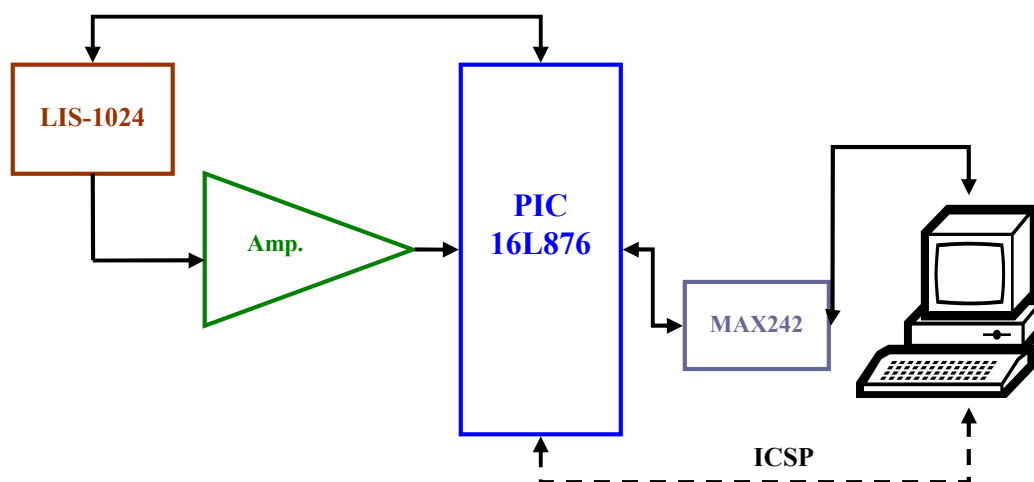


Fig. 4.1 – Diagrama de blocos do sistema *autónomo*

Assim, em síntese, os dados da imagem são colectados num sensor do tipo matriz linear de 1024 *pixels* sendo todo o processo de aquisição, transferência de dados e todas as funções do sensor linear sincronizadas pelo *PIC*. É o microcontrolador que controla a leitura do sensor assim como a frequência de leitura ou a paragem da mesma. Aqui a frequência do relógio/taxa de leitura dos *pixels* corresponde a um período de 888 μ s. A programação do *PIC*, o *software* de aquisição e tratamento de dados no *PC* será objecto de estudo na fase seguinte deste capítulo. Os dados recebidos pelo sensor são enviados para o *PIC* mas têm primeiro de ser amplificados (pelo amplificador *TL081CPN*). Este amplificador está vocacionado essencialmente para a utilização em dados provenientes de aplicações vídeo. O *PIC* recebe estes dados analógicos num *ADC* interno que os converte em digitais e transfere-os, usando a sua *USART*, para a *MAX242* que irá comunicar com o *PC* pelo protocolo *RS232*. O *driver* *MAX242* faz a conversão dos sinais digitais para a sinalização da norma *RS232*, usada na comunicação com o porto série do *PC*. Permite receber ordens do *PC* para dar início à leitura do *PIC* ou escrita deste na *MAX242* sem haver interferências de dados. O *PIC* poderá ser programado do exterior pelo conector *IDC6M* (*ICPS* – *in circuit serial programming*)

como se vê no esquema da figura 4.1. Essa programação será realizada com o pacote *MPLAB* (*Microchip*). Tal como já se referiu anteriormente, um dos aspectos que levou a optar por esta arquitectura foi o conhecimento da ferramenta *MPLAB* da *Microchip*. O *PCB* pode ser visto na figura 4.2, assim como os seus componentes electrónicos. O desenho respectivo encontra-se no apêndice B.

Para melhor compreensão do circuito electrónico vamos descrever os seus componentes principais em maior detalhe. Para auxiliar recomenda-se a consulta do esquema completo do circuito incluído como apêndice A.

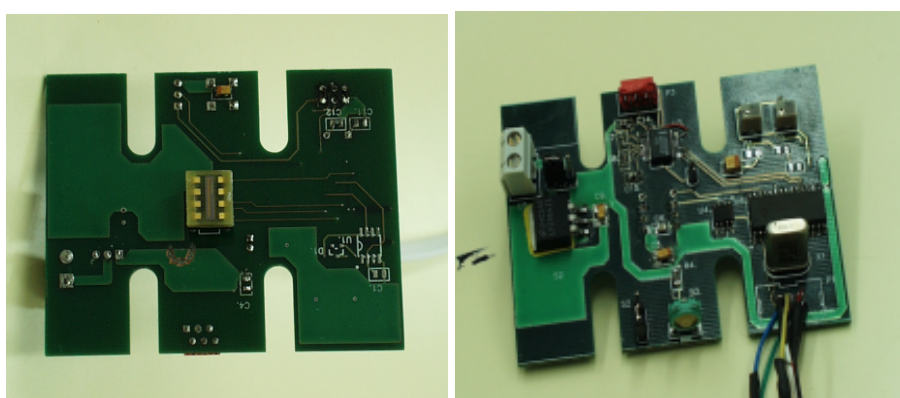


Fig. 4.2 – PCB introduzido no circuito óptico: i) face que contém o sensor; ii) face oposta onde estão os restantes componentes electrónicos

4.2.1. Sensor *CMOS LIS-1024*

O sensor *CMOS LIS-1024* é um sensor de imagem, linear, de gama superior da *Photon Vision Systems* e as suas funcionalidades têm variadíssimas aplicações em áreas como a medicina, automação industrial etc. O sensor consiste numa matriz de 1x1024 *pixels fotodíodos*, com um encapsulamento de oito pinos e as suas principais propriedades são: baixo custo; operação única de alimentação; *reset* dinâmico dos *pixels*; grande sensibilidade; razão *SNR* elevada; capacidade de fazer leituras não destrutivas; operação na gama de 1kHz a 40MHz; baixa corrente negra (*dark current*); integração de funções de temporização e de controlo; permite substituir sistemas *CCD* e não apenas os sensores. O dispositivo oferece múltiplos modos de operação tais como: leitura sequencial destrutiva ou não destrutiva e *frame* de leitura não destrutiva ou

destrutiva. No modo sequencial (síncrono), designado por modo *DPR* (*Dynamic Pixel Reset*), permite-se que cada *pixel* seja apagado assim que é lido garantindo que cada *pixel* seja integrado no mesmo intervalo de tempo, fazendo com que este seja continuamente iluminado como uma aplicação de prova. Pode também efectuar a leitura dos *pixels* sem reiniciar os seus valores, o que se designa por leitura não destrutiva.

O *clock* (*CLK*) é fornecido mediante a taxa desejada de leitura dos *pixels*, escolhendo o modo de *reset* para iniciar a leitura externa quando corre assincronamente. Assim no modo *frame* o ciclo de leitura é feito pulsando o pino *RST*. Escolhe-se o nível utilizado com o *PRE* (activação do *reset* do *pixel*) e *RMS* (escolha do modo de *reset*). Neste modo o fim de cada *frame* é identificado com uma saída de um impulso do pino *SYN*. O impulso *SYN* começará na parte ascendente do ciclo de *clock* $1024+2$ (1024 é o número de *pixels*) para aquele *frame*. O impulso *SYN* tem a sua descida no impulso $1024+4$. O *Reset* do contador interno é no limite ascendente do *clock* ao impulso $1024+5$. O relógio tem de percorrer $1024+5$ ciclos para contagens internas até serem apagados. O relógio pode continuar ou parar passados os $1024+5$ ciclos. O tempo de integração total é de 1029 ciclos (ou 1029 períodos do relógio).

O sensor opera bem até à velocidade de 12MHz no modo de operação *DPR*. Acima deste valor não é recomendado sendo preferível a escolha do modo *frame* já que o último *pixel* no modo *DPR* é inválido a menos que a extremidade da *frame* seja restaurada comutando os modos. Neste modo o impulso *RST* restaura o contador interno e a saída de vídeo é preparada para ler o primeiro *pixel*. Devido à linha *PRE* estar desactiva os *pixels* não são restaurados e a imagem é preservada.

O modo *frame* de leitura destrutiva, é conseguida da mesma maneira que o outro modo *frame* de leitura não destrutiva só que quando o sinal *PRE* está activo este vai causar o restauro dos *pixels* quando o sinal *RST* é activado.

4.2.2. PIC16F876™

O *PIC* ou *Peripheral Interface Controller* (controlador de interface periférico) é uma tecnologia criada pela *Microchip Technology Inc.*™, e é a designação

para identificar os seus microcontroladores montados num único *chip*. Tem uma grande procura no mercado o que resulta das suas características de entre as quais, para o caso do *PIC16F876* se passam a resumir as principais: 368 *bytes* de memória *RAM*; 8 *kbytes* de memória de programa; 22 pinos de entrada/saída (configuráveis por *software*); três contadores (*timers*); duas portas série; duas comunicações em série (modo *USART* e *MSSP*); suporte *SPI* (interface periférica de série) e bus *I²C* (circuito integrado); *ADC* interno de 10 bits com 5 canais de entrada; frequência máxima de relógio de 20MHz. Estas características permitem que se reduzam componentes externos, aumentando assim a fiabilidade e a previsibilidade de todo o sistema.

O programa e dados são acedidos em diferentes memórias através de *bus* diferentes o que permite aumentar a largura de banda. Consegue-se assim que as instruções tenham tamanhos diferentes das palavras de dados de 8 bits de largura.

A família *PIC 16F87X* é acompanhada por um poderoso compilador. Este compilador simplifica o processo de desenvolvimento e teste de uma dada aplicação. O compilador possui a possibilidade de “traduzir” código *C* para *Assembler*. Esta característica aliada ao facto de haver garantia que todas as instruções são executadas a uma frequência fixa determinada ($\frac{1}{4}$ da frequência máxima ou seja 20MHz/4 o que resulta em 200ns por instrução) torna o *PIC* um componente muito desejado, pois permite que o tempo de execução de cada rotina seja controlada facilmente e com grande segurança

Outras características igualmente importantes mas de índole interna são:

- Controlo de interrupção – a *CPU* pode controlar treze fontes de interrupção independentes;

- O *chip* só funciona dentro da tensão de especificação – caso contrário deixa de funcionar (*power-on reset*, *brown-out* e *watchdog timer* para protecção);

- Modo *SLEEP* – permite reduzir o consumo de energia.

De entre as características periféricas já referidas deve-se salientar, devido à sua importância para este trabalho, a existência de três *Timers*: *Timer 0*:- *timer* de 8 bits com *prescaler* de 8 bits; *Timer 1*: *timer* de 16 bits com *prescaler* que pode ser incrementado durante o *sleep* através do cristal externo; *Timer 2*: *timer* de 8 bits com

registo de período de 8 bits, *prescaler* e *postcaler* programáveis e com registo de períodos de 8 bits¹.

Uma das potencialidades dos *timers* do *PIC* é a sua funcionalidade de definir o *PWM* que permite criar uma onda quadrada. Esta onda terá as partes que a constituem diferentes caso seja necessário. Os resultados que se obtêm e interessam são:

PWM período (μ s)	PWM duty cycle (μ s)	Frequência (kHz)	CCPR1L	PR2
888	444	1.1	128/0x80	255/0xFF

Esta é a temporização que nos interessa para os cálculos. Sendo compatível com o tempo de conversão e aquisição do *ADC*.

Os registos mais importantes são: *PR2*; *CCPR1L*; *CCP1CON* (5:4); *T2CON*.

A programação do PWM:

- ▶ *Escolher o período de PWM definindo o registo PR2;*
- ▶ *Escolher o duty cycle de PWM definindo os registos CCPR1L e CCP1CON<5:4>;*
- ▶ *Colocar o pino CCPI como saída limpando o bit TRISC<2>;*
- ▶ *Colocar o valor do prescaler TMR2 e activar o Timer2 definindo o registo T2CON;*
- ▶ *Configurar o módulo CCP1 para operar em modo PWM.*

Outro dos componentes importantes do *PIC* é o *ADC* que é indispensável para esta aplicação, pois faz a conversão da informação analógica proveniente do sensor para digital que poderá ser enviada para o *PC*.

Para a programação do módulo *ADC* existem quatro registos importantes (*ADRESH*, *ADRESL*, *ADCON0* e *ADCON1*) que se passam a descrever sucintamente: *ADCON0* (*A/D control register0*) controla as operações do módulo *A/D*; *ADCON1* (*A/D control register1*) configura as funções dos pinos do porto; *ADRESH/ADRESL*

¹ Para mais informação acerca do *PIC*, como a sua arquitectura, memória, descrição de cada pino pode ser consultada a sua data sheet no site <http://www.microchip.com>.

(*A/D result high register/A/D result low register*) contêm o resultado em 10 bits da conversão *A/D*. Quando a conversão está completa, o resultado é carregado neste par de registos *A/D*, o bit *GO/DONE* (*ADCON0<2>*) é limpo e a *flag* de interrupção é activada.

A programação do ADC (configuração e leitura):

- 1) Configurar o módulo A/D: configuração dos pinos analógicos, tensão de referência e I/O digitais (ADCON0); escolha do canal de entrada A/D (ADCON0); escolha do clock de conversão A/D (ADCON0); colocar o módulo activo A/D (ADCON0);*
- 2) Configurar a interrupção A/D se necessária: limpar ADIF; colocar ADIE; colocar GIE;*
- 3) Esperar pelo tempo de aquisição;*
- 4) Começar a conversão: Colocar o bit GO/DONE (ADCON0);*
- 5) Esperar pela conversão completa, por um ou outro caso: polling do bit GO/DONE para ser limpo ou esperar pela interrupção do A/D;*
- 6) Ler os resultados do par de registos A/D (ADRESH/ADRESL), limpar o bit ADIF se necessário;*
- 7) Para uma próxima conversão voltar aos passos 1) e 2). O tempo de conversão por bit é TAD com o valor mínimo de 1,6µs; Tem que se esperar 2 TAD antes da próxima conversão. O tempo mínimo de aquisição é de aproximadamente 20µs.*

A configuração, em termos de *software* do *PIC* poderá ser consultada nos apêndices correspondentes e a sua programação será abordada posteriormente.

Apresentam-se em seguida as opções tomadas para a configuração dos pinos *I/O* do *PIC*, como vemos no apêndice A. Alguns desses portos são multiplexados, com uma função alternativa, para funções periféricas do dispositivo. Em geral, quando um periférico está operacional, esse pino não pode ser usado como *I/O*.

O porto A do *PIC* é uma *latch* de 6 bits. A opção foi a de ligar a este porto as linhas das tensões de referência (V_{ref+} e V_{ref-}) e a entrada de dados analógicos

provenientes do sensor por intermédio do amplificador *TL081CPN*. Todos os pinos deste porto têm níveis de entrada *TTL* e de saída *CMOS*. Podem-se configurar os pinos deste porto como entradas ou saídas por *software*. Temos como opção colocar no terceiro estado de alta impedância.

O porto B é um porto bidireccional de 8 bits em que 4 podem provocar interrupções. Neste porto usam-se alguns pinos para programar o *PIC* do exterior através das ferramentas de *software* e *hardware* da *Microchip* (*MPLAB*) por via da ficha *IDC6M* e outros para as linhas de controlo da leitura do sensor linear.

O porto C é também um porto bidireccional de 8 bits. Também aqui os pinos são configuráveis individualmente, por *software* ou por registos internos. A este porto estão ligadas as linhas de transmissão e recepção de dados pelo *PC* via porto série e a linha de *clock* do sensor que estabelece a sua taxa de leitura.

Os pinos 9 e 10 são pinos de entrada para o oscilador interno do *PIC*, ou *clock* do *PIC*. Este *clock* pode ir até um máximo de 20MHz e tem ciclos de instruções até 200ns. É através destes pinos que temos o gerador de tempo real e está ligado directamente ao *watchdog timer*. Este *timer* faz o *reset* ao *PIC* se forem detectados erros de funcionamento no *chip*, ou se houver uma alteração nas suas operações normais.

É através do pino 1 (*MCLR*) que se faz o *reset* ao *PIC* quando este recebe energia pela primeira vez da alimentação ou quando é activado do exterior. O diodo *DI*, que se vê no regulador de tensão, é usado para prevenir possíveis problemas com a corrente nessa entrada do *PIC*. Quando a tensão tiver um certo valor o diodo satura impedindo a passagem de corrente evitando deste modo o perigo de subcarga do *PIC*.

Para terminar, uma referência ao que distingue um microcontrolador dos outros processadores e que é o facto de consistirem de circuitos especiais para lidar com necessidades de aplicações em tempo real. O *PIC* tem várias características para maximizar a fiabilidade do sistema, minimizar custos através de eliminação de componentes externos, fornecer modos de operação com poupança de energia e oferecer protecção do código.

4.2.3. Amplificador TL081CPN

Trata-se de um amplificador operacional (da *SGS-Thomson Microelectronics*) utilizado essencialmente em aplicações de electrónica comum.

As suas principais características são: baixa potência consumida; baixa corrente de entrada de *offset*; *slew rate* elevado ($16\text{V}/\mu\text{s}$); frequência interna de compensação; protecção de saída de curto-circuito; elevada tensão de rejeição de modo comum (no máximo 18V). Podem-se ver em detalhe as ligações entre o microcontrolador, o sensor e o amplificador na figura 4.3.

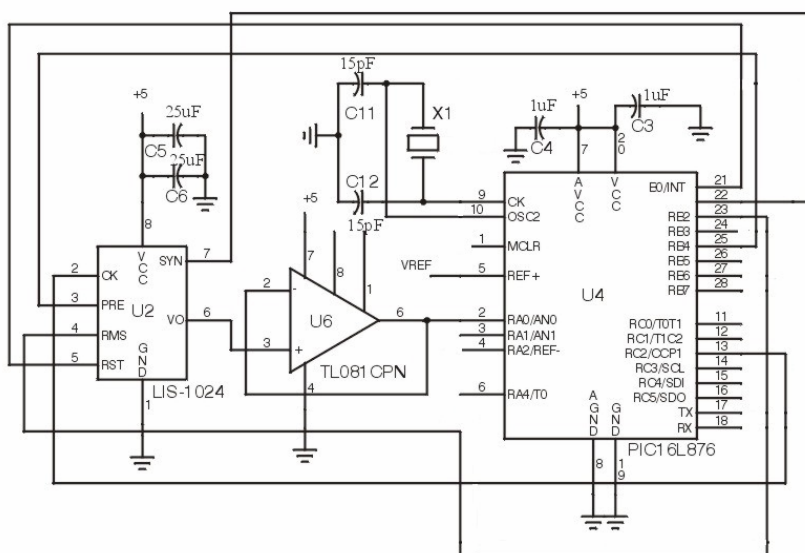


Figura 4.3 – Ligações no *PCB* entre o sensor, o *PIC* e o amplificador

4.2.4. Protocolo RS232

A maioria dos *PC* tem dois portos série, usando o protocolo *RS232*, que são vulgarmente designados de *COM1* e *COM2*. Estes portos podem ser usados para comunicar com outros periféricos como o rato ou modem. Frequentemente o *PC* tem endereços *standards* para dois portos adicionais (*COM3* e *COM4*). As velocidades de transferência de dados nesta interface são limitadas, normalmente iguais ou menores a 256 Kbps, tal como o comprimento do cabo, definido inicialmente com o valor máximo

de 50m. No entanto, actualmente vemos portos, no nosso *PC*, com grande velocidade e cabos com maior comprimento sem que ocorram falhas.

Este protocolo é para comunicações de dados electrónicos *standard*, criado pela *EIA* (*Electric Industry Association*). A sigla *RS* é *Recommended Standard*.

Os dados electrónicos são divididos em duas categorias, ou seja são diferenciados, e designados de *single-ended*. O protocolo *RS232* gere este tipo de comunicação *single-ended* em que existem canais independentes que permitem estabelecer dois sentidos de comunicação.

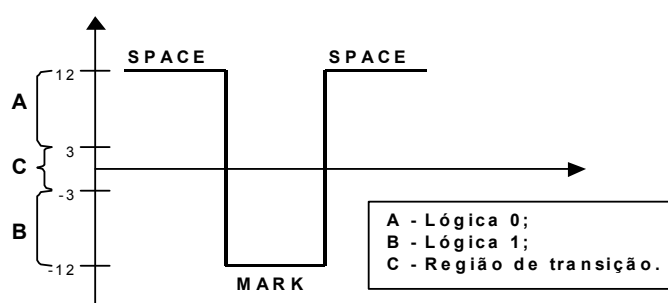


Figura 4.4. – Sinais de saída do RS232.

Os sinais *RS232* são representados por níveis de tensão em relação a um nível de referência comum (*power /logic ground*). O estado “preguiçoso”, de -3V a -12V, (*MARK*) tem um nível de sinal positivo em relação ao sistema comum. O sinal de saída varia entre 12V e -12V, tendo uma zona morta entre 3V e -3V.

Pino	Sinal	Descrição
1	CD	Detecção de recepção
2	RD	Recepção de dados
3	TD	Transmissão de dados
4	DTR	Recepção de dados pronto
5	SG	Sinal de terra
6	DSR	Envio de dados pronto
7	RST	Limpar para enviar dados
8	CTS	Pedido para enviar dados
9	RI	Indicador de dados

Tabela 4.1. – Designação e função de cada pino da ficha *RS232*.

O *RS232* pode usar diversas linhas de comunicação, que são específicas deste protocolo, para evitar a ocorrência de erros. Na tabela 4.1. mostra-se a descrição, e função de cada pino, das fichas *RS232*.

A transição numa ficha de 9 pinos é feita por: 1→7-8 (alternando uma com a outra caso de pedido de envio de dados ou limpar dados); 2→ 3; 3→ 2; 4→ 6; 7-8(alternando uma com a outra caso de pedido de envio de dados ou limpar dados) → 1. Na figura 4.5., apresenta-se o esquema das fichas *D9* normalmente usadas para implementar o protocolo *RS232*, tanto a ficha macho como a fêmea, com a indicação dos pinos que constam da tabela 4.1.

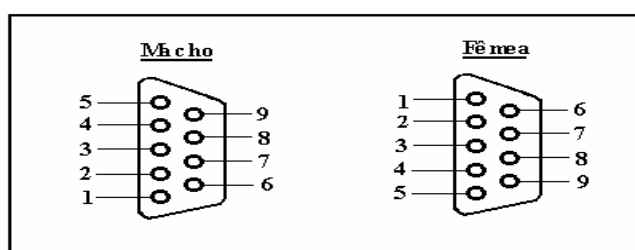


Figura 4.5. – Fichas de ligação *RS232*.

4.2.4.1. MAX242

As principais características do *MAX242* da *Maxim* são: modo de recepção de baixo consumo; activação e desactivação separados; tem linhas de transmissão, controlo, emissão e recepção; recepção e transmissão múltiplas, com saídas no terceiro estado; saída de recepção, com entrada de indicação de estar em condição para tal (*EN*).

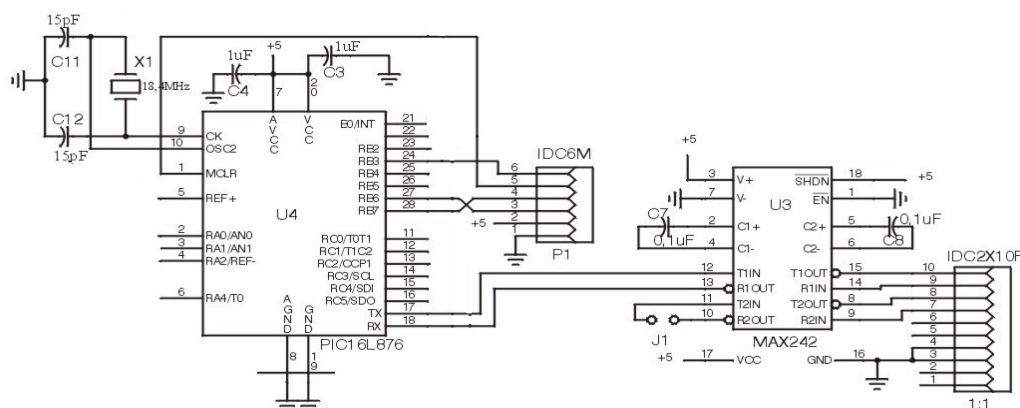


Fig. 4.6 – Ligações no *PCB* entre o *PIC*, a *MAX242*, o *IDC6M* e a ficha de comunicação com o *PC*

As ligações entre o microcontrolador e o *MAX242* que estabelece as comunicações do *PCB* com o *PC* podem ver-se em detalhe na figura 4.6, em que se mostram também os conectores *IDC6M*, para ligação ao *PC* para programação *online* do *PIC*, e o *IDC2X10F* que recebe os sinais RS232 de saída do *MAX242* para ligação ao porto série do *PC*.

4.3. Programação do PIC

A programação do *PIC16F876* foi desenvolvida na linguagem *C* da *Borland* usando o *MPLAB*, uma ferramenta de desenvolvimento de *software* da *Microchip*. A placa é conectada através de uma aplicação de interface específica com o módulo *MPLAB-IDC* que é conectado com o *PC* por uma porta série. Na placa existe um *jumper* para seleccionar o modo *run* para funcionar normalmente na aquisição de dados (quando não está conectado ao *MPLAB*) ou *debug* para uma tarefa de desenvolvimento (quando conectada ao módulo mencionado). O programa está contido no ficheiro *Sensor5.c* que está disponível no apêndice D. A definição do modo de leitura do sensor e sua comunicação com o *PIC*, assim como a inicialização e configuração dos diversos periféricos do *PIC* é aqui realizada. O programa principal consiste num ciclo infinito usando interrupções capazes de activar *flags* e seleccionar rotinas para serem executadas. O início dá-se com a recepção de um carácter pré-definido, pela *USART*, introduzindo pelo utilizador no teclado do *PC* e é desactivada a *flag rxflg*. Se houver já uma aquisição em curso o microcontrolador dá ordem para o término desta, caso contrario dá-se início á aquisição dos dados no sensor. A leitura do sensor começa quando recebe a respectiva ordem do utilizador que ocorre quando este pressiona a tecla 'i' ou 'I' no teclado do *PC* e termina da mesma forma, isto é, com outra ordem recebida do utilizador, neste caso pressionando qualquer tecla.

O programa que corre no *PIC* consiste num ciclo infinito, como já foi referido, mas existe um conjunto de rotinas de interrupção para, entre outras, verificar se recebe algum carácter por parte do utilizador. No caso de receber um carácter do utilizador então a *flag rxflg* é activada na rotina de interrupção e no ciclo infinito é verificado se corresponde a uma ordem válida do utilizador. Em caso afirmativo dá

início à leitura (activação da *flag iniflg* e desactivação da *flag rxflg*). Portanto só quando a *flag iniflg* está activa é que se dá início ao processo que leva à leitura do sensor. Em seguida a *flag iniflg* é desactivada sendo activado o *PWM*, e o respectivo *duty cycle*. Dá-se assim início ao ciclo de leitura do sensor, propriamente dito, sendo activados o *ADC* e o *Timer0* (define o tempo de conversão do *ADC*) do *PIC* para ler e converter os dados provenientes do sensor. Também é necessário activar o *Timer1* que é usado para definir os instantes de leitura do sensor em que para cumprir as especificações do sensor é feito um atraso para que o início da aquisição coincida com o meio do ciclo de relógio de leitura do sensor (definido pelo sinal de saída do *PWM*) garantindo uma aquisição correcta dos dados. Assim que o *ADC* termine a aquisição do dado, e respectiva conversão, é activada a *flag adcflg*. Para cada leitura de um *pixel*, aproveitando o tempo de espera necessário para a conversão do *ADC*, o valor do *pixel* precedente é carregado no *buffer USART* para ser então enviado para o *PC*. O ciclo de leitura termina quando todos os *pixels* do sensor forem lidos ($naquis = NPIXEL$). A sincronização necessária para a leitura do sensor é determinada no fim de cada *frame*. Depois da conclusão da leitura do último *pixel* o programa espera pelo impulso *SYN*, de saída do sensor, para reiniciar o processo de leitura. O diagrama de blocos da leitura pode ser visto na figura 4.7.

Na definição deste processo teve-se em atenção as temporizações de modo a não haver perda de informação, de acordo com as especificações do sensor, e otimizar algum do tempo de espera que decorre das temporizações dos diferentes processos e das respectivas interrupções. O período do *PWM* (é o relógio do sistema) é de $880\mu s$ mas devido á imposição do sensor o *Timer1*, que vai ter o mesmo período do *PWM*, vai iniciar a sua interrupção com um atraso de meio período ($iniTimer1\ delay = 440\mu s$). Inicia-se a contagem com o *Timer0*, cujo período é o tempo de aquisição do *ADC* ($Tacq$), e esse tempo de aquisição tem o valor de $30\mu s$. Em seguida há a ordem de conversão do *ADC* (activa-se a interrupção do *ADC*) cujo tempo de conversão é de $25\mu s$ ($Tconv$). A figura 4.8 representa estas temporizações aqui explicadas. A frequência de relógio, ou taxa de leitura dos *pixels*, é baixa (1-10kHz) devido ao limite imposto pela velocidade de comunicação (designada por *baud rate*) do *RS232* (9600-57600bps).

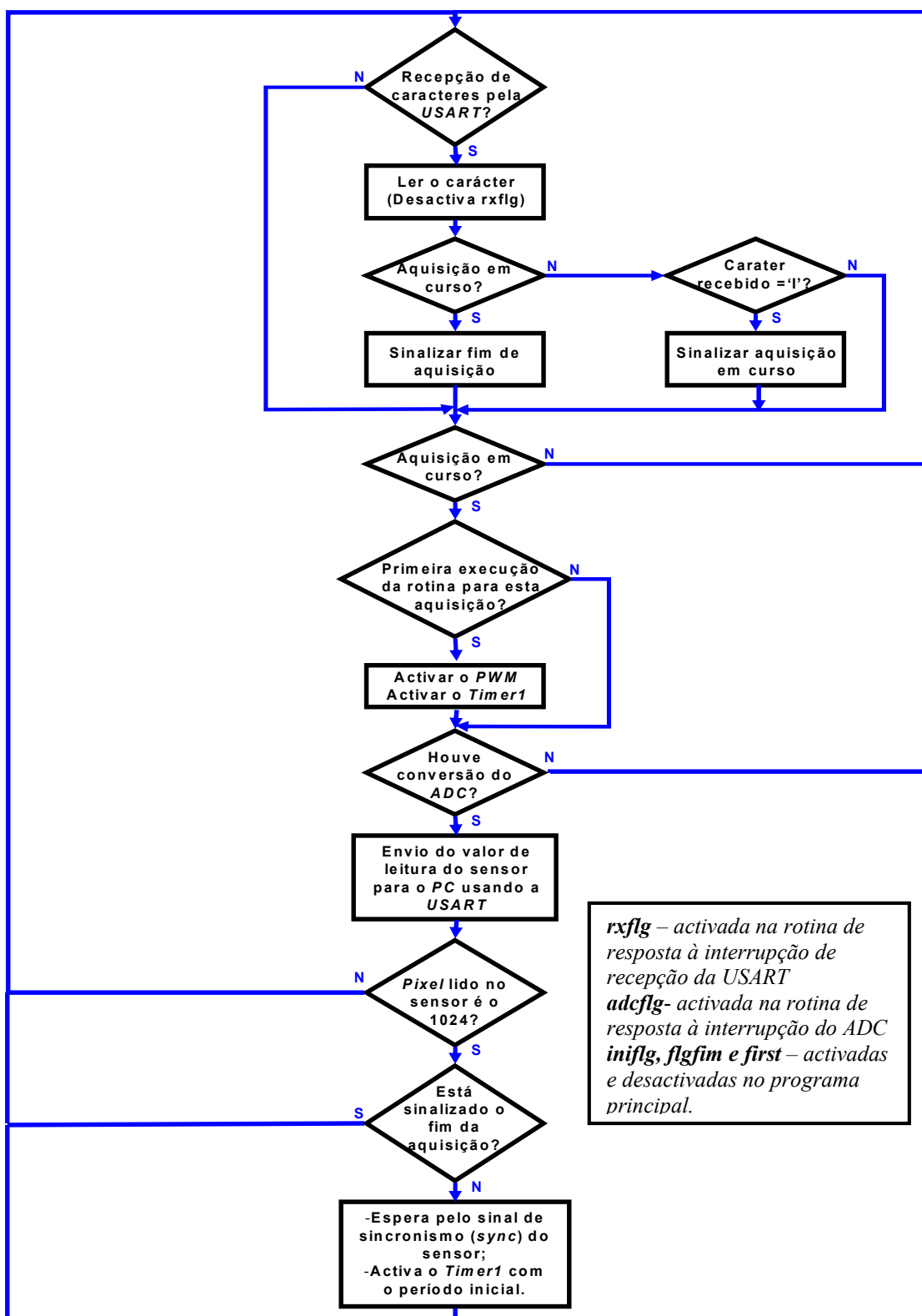


Fig. 4.7 – Diagrama do processo de inicialização e leitura do sensor.

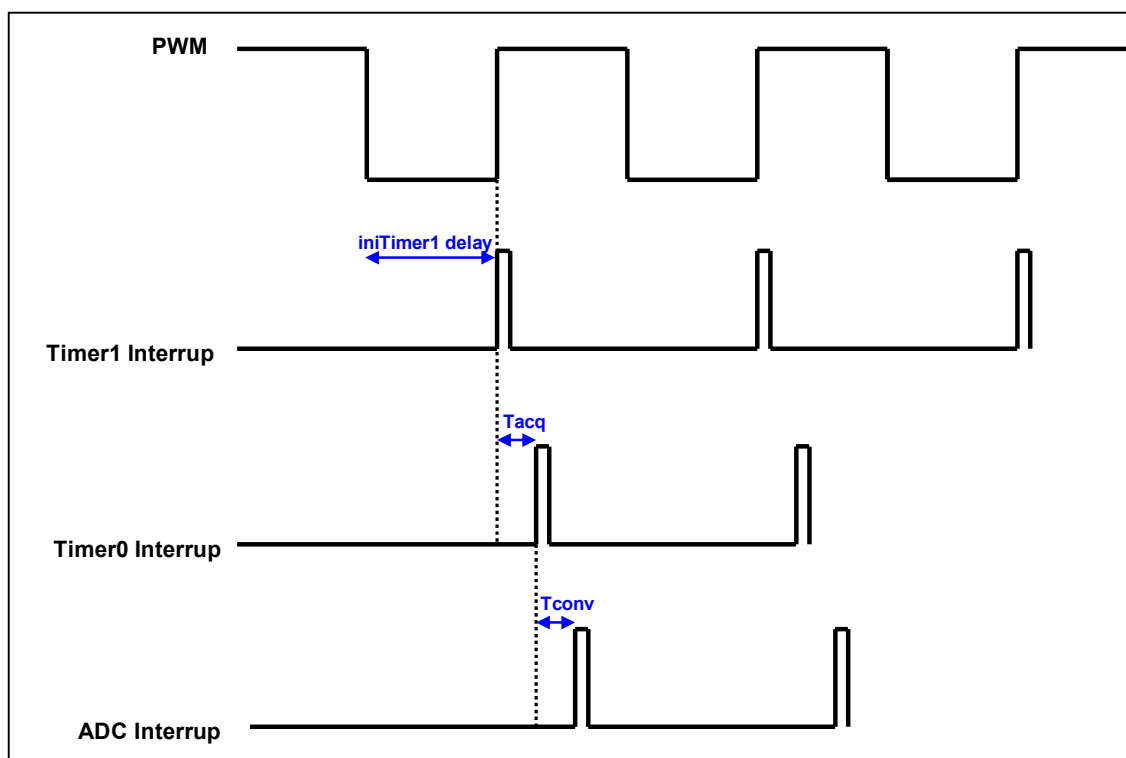


Figura 4.8 – Temporização dos componentes do circuito (note-se que os intervalos de tempo das interrupções estão amplificados para melhor visualização)

Com a ajuda da figura 4.9 vamos descrever esse processo. A comunicação entre o *PC* e o *PIC* é feito com uma palavra de três bytes. O primeiro byte é composto por 6 bits iniciais, que servem para definir o cabeçalho da mensagem e os outros dois bits finais, correspondem aos bits mais significativos do valor de 10 bits obtido pelo *ADC* que limitam o primeiro byte. O segundo byte é composto pelos restantes 8 bits (menos significativos) do *ADC* do valor de cada um dos 1024 *pixels* do sensor lidos. O último byte é o término da palavra, onde se verifica se toda a informação foi enviada sem erros.

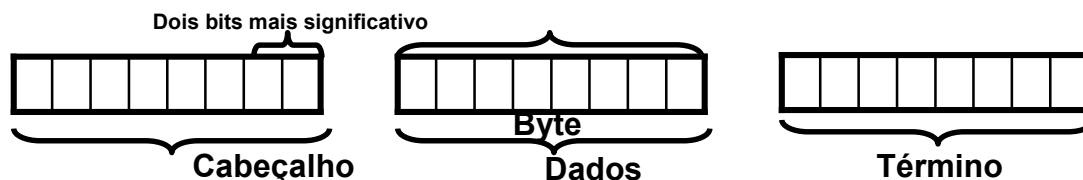


Figura 4.9 – Transferência de dados entre o *PIC* e o *PC*

4.4. Programa de visualização dos dados no PC

A programação no *PC* foi desenvolvida na linguagem *C* da *Borland*. A sua estrutura assenta nos seguintes ficheiros:

- Confocal.h;
- Conf-com.c;
- Confocal.c.

No *header Confocal.h*, que pode ser visto no apêndice *C*, inicializam-se definições genéricas, funções específicas, variáveis globais, ficheiros de configuração, mensagens, erros de transmissão e recepção, o *buffer* de comunicação e identificação das portas de comunicação do *PIC* e a configuração das comunicações com a porta série *RS232*.

No ficheiro que se encontra no apêndice *E*, o *Conf-com.c*, está definida a configuração das comunicações série entre o computador e o *PIC*, assim como das portas série utilizadas. Foram definidas mensagens usadas na comunicação e comandos referentes às portas série. Podemos ver no apêndice *E* as rotinas de inicialização da comunicação série, da reposição do estado inicial do computador, rotinas de interrupção para transmissão por parte do utilizador e da recepção de caracteres pela porta série assim como rotinas de envio de mensagens com um determinado tempo de espera.

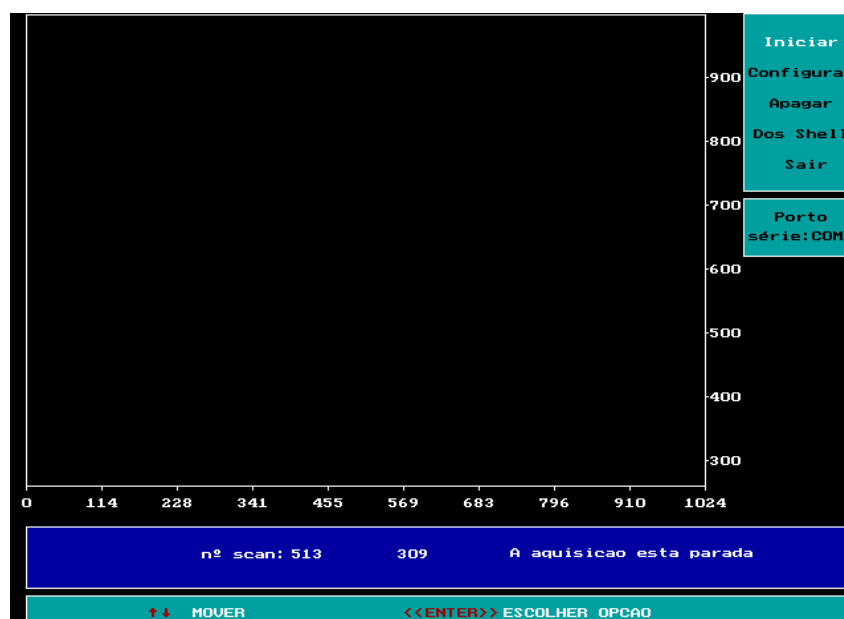


Figura 4.10 – Interface com o utilizador do programa Confocal.c

Por fim tem-se no apêndice F o programa *Confocal.c* que contém o código desenvolvido para a visualização gráfica no computador dos resultados lidos do sensor. Além de diversas funções para a construção da interface com o utilizador contém então a função que permite mostrar os valores lidos do sensor sob a forma de gráfico em que é chamada a função de recepção pelo porto série. Para dar início á aquisição de dados do sensor selecciona-se a opção *Iniciar* do *Menu* à direita que usa uma função de envio de um caracter previamente definido através do porto série para o *PIC* no módulo de leitura do sensor. Na figura 4.9 apresenta-se a interface com o utilizador deste programa de visualização de dados.

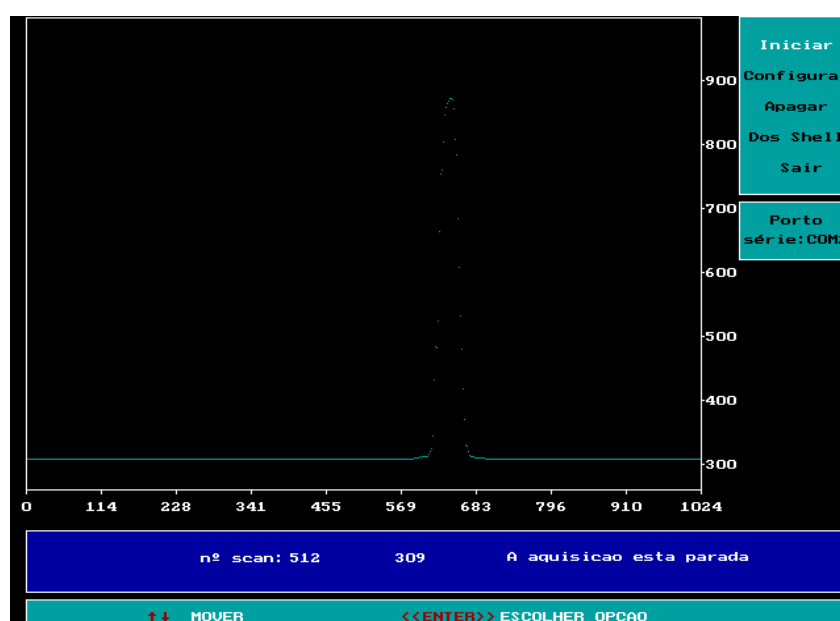


Figura 4.11 – Resultados obtidos no sensor com utilização de uma fenda.

Como já foi referido, definimos um menu principal, no canto superior direito, com várias opções como vemos na figura 4.10. Existe também a possibilidade de seleccionar a respectiva porta de comunicação utilizada para o efeito. De entre as funções para definir a *interface* com o utilizador constam a função para definição da janela principal do programa e naturalmente a função para a definição do menu que vêm descritas no apêndice F. Também se podem analisar os pontos obtidos no gráfico segundo o eixo das abcissas assim como a sua ampliação. Como funções na opção *Configurar* tem-se: alterar a escala, visualizar os valores médios ou uma escala de valores diferente da normal e escolher a *sample rate* entre outras opções. Para ilustrar a

visualização de dados na figura 4.11 apresentam-se os valores dos 1024 *pixels* lidos do sensor, no caso usando uma fenda no caminho óptico de iluminação feita com um laser.

Em suma, este programa foi utilizado como uma primeira plataforma para visualização dos primeiros resultados do sistema óptico inicialmente proposto, o que permitiu realizar também o teste e verificação do correcto funcionamento do sensor na placa de aquisição de dados em simultâneo com o teste das comunicações.

5. Resultados e Discussão

5.1. Introdução

O arranjo óptico usado neste trabalho consiste num protótipo laboratorial de um microscópio de varrimento do objecto no modo de reflexão epi-iluminada^[57, 58]. Assim não é efectuado qualquer movimento dos componentes ópticos e as lentes são usadas na sua parte central para minimizar as aberrações. O varrimento mecânico na direcção paralela ao sensor linear normalmente não é necessário apesar de também ser possível. Como objectivo final do projecto em que este trabalho se inclui, pretende-se que o sistema óptico não utilize *pinholes* mas unicamente use a informação proveniente do *CCD* linear em algoritmos de restauro (*restoration*) ou algoritmos de desconvolução^[55].

A contribuição deste trabalho é o de avaliar a resposta axial do sistema para diferentes configurações da montagem óptica. Para isso ter-se-á que garantir que se obtém resultados consistentes com diferentes sistemas ópticos. Para tal teve-se o máximo cuidado na sua utilização de forma a não ser necessário efectuar alinhamentos ópticos sempre que se pretendia usar o sistema ou seleccionar outra combinação de componentes ópticos.

5.2. Montagem óptica

A montagem óptica consiste numa configuração epi-iluminada no modo de reflexão tendo sido testadas várias opções: Estas podem ser resumidas nos 3 grupos seguintes:

- 1) Iluminação total ou em linha;
- 2) Campo amplo ou com *pinhole* de detecção (com diferentes diâmetros);
- 3) Lente objectiva asférica ou acromática.

A configuração base é a apresentada na figura 5.1, sem incluir os componentes envolvidos por uma oval a tracejado. As diferentes configurações utilizadas resultam da introdução na montagem de um ou mais desses componentes, como por exemplo a colocação de um *pinhole* ou de uma fenda ajustável^[57]. Esta fenda de largura variável é também utilizada como filtro espacial, sendo útil também para eliminar aberrações que podem prejudicar a resolução (axial e lateral). Por outro lado tem que se encontrar a combinação adequada entre a intensidade da luz utilizada e a gama dinâmica do sensor para não criar interferências que possam provocar saturação ou dispersão da luz no sensor^[2,55].

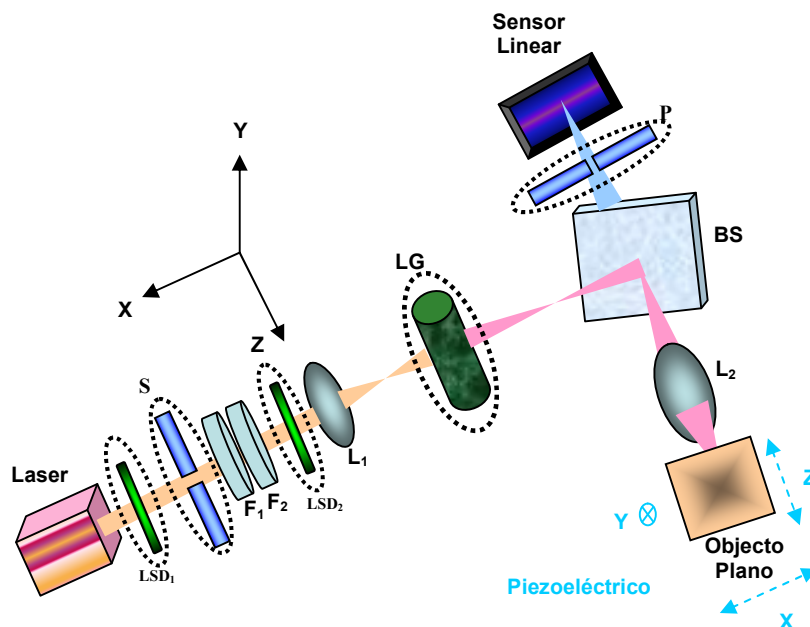


Fig. 5.1 – Esquema da montagem óptica

O sistema óptico, que foi montado numa mesa *Melles Griot*, tem componentes mecânicos na sua maioria do sistema *Microbench* da *Linos (Spindler & Hoyer)*^[57,58]. Contém os seguintes componentes ópticos e mecânicos:

- **Laser** – Díodo laser vermelho de 5mW ($\lambda=635\text{nm}$) com regulador de intensidade;
- **LSD₁** e **LSD₂** – *Light Shaping Diffuser*, é um registo holográfico constituído por estruturas aleatórias de superfície em relevo com grande eficiência de transmissão (para cima de 92%), apresentam uma distribuição angular controlada que fornece uma

grande qualidade de luz homogeneizada. Estas estruturas totalmente aleatórias (não periódicas) não dependem do λ e eliminam o efeito de *Moiré* sem aberrações cromáticas. A precisão das estruturas de superfície em relevo fornece um controlo da divergência de luz angular, imitando o funcionamento de uma lente negativa. São produzidos em vários modelos standard, ou específicos, com grande variedade de ângulos, tamanhos, forma, cores, com diferentes cortes para diferentes peças e com diferentes materiais do substrato incluindo dois lados difusores;

- **S** – Fenda ajustável com largura variável entre 0mm e 4mm com um *driver* micrómetro de $2\mu\text{m}$ de resolução (*Melles Griot 07SLT001*);

- **F₁** e **F₂** – Filtros de polarização ou ND (*Neutral Density*) que são usados para adequar a intensidade da luz de iluminação à gama dinâmica do sensor;

- **L₁** – Lente plano-convexa (ou esférica) com $f=20\text{mm}$ que actua como lente colectora e que permite obter o feixe com divergência adequada;

- **LG** – *Line Generator* que é uma lente anamórfica que permite obter um feixe planar que se vai situar no plano horizontal (*X-Z*);

- **BS** – *Beam Splitter* (divisor de feixe) que vai reflectir o feixe de iluminação para a lente objectiva e simultaneamente transmitir a luz proveniente do objecto. Tem baixos índices de absorção (dimensões: 20x30mm; espessura=2,5mm; $n=1,515$);

- **L₂** – Lente objectiva do sistema tendo sido usadas duas lentes de baixo custo: lente acromática (*Achromat Doublet 5x*, 0.42 NA) e lente esférica (*Aspheric 11x*, 0.57 NA). O comprimento do tubo é ajustável por isso a ampliação é variável;

- **P** – *Pinhole* usado como filtro espacial dos raios provenientes do sensor (*pinhole* de detecção);

- **Objecto Plano** – Como objecto usou-se um espelho dieléctrico *broadband* plano (pode-se indicar como a nossa referência) que está colocado numa plataforma de translação para medir a resolução axial;

- **Plataforma de translação** – Permite efectuar o posicionamento do objecto através de um parafuso micrométrico com uma resolução de $2\mu\text{m}$, ao longo dos três eixos, como vemos na figura 5.1.

- **Sensor Linear** – Está integrado na placa de circuito impresso (descrito no capítulo anterior).

Como já foi referido atrás os componentes ópticos envolvidos por uma oval a tracejado na fig. 5.1 são opcionais, isto é, foram usados em alternativa podendo ser usado mais do que um em simultâneo mas não todos no mesmo sistema. Foram assim testados diferentes configurações do sistema óptico, cujas variações são definidas com base neste conjunto de componentes ópticos, tendo-se obtido resultados com algumas diferenças que serão apresentados e discutidos neste capítulo.

Para ser possível obter, com este sistema óptico, resultados que permitam avaliar a sua confocalidade bem como medir a sua resolução lateral e axial, é imprescindível realizar os varrimentos, isto é, os movimentos da plataforma de translação tendo-se usado os parafusos micrométricos. Nesta fase este procedimento manual não é desadequado mas no âmbito do projecto em que este trabalho foi desenvolvido foram posteriormente incluídos no sistema três actuadores que constituem um sistema de varrimento que permite recolher, de forma automática, luz de diferentes planos do objecto com distâncias pequenas ^[57,58].

O sistema óptico montado na mesa óptica, ou seja, o protótipo do microscópio pode ser visto na figura 5.2, que ilustra também a forma como a placa de *PCB* desenvolvida para leitura do sensor *CMOS* linear é integrada no sistema.

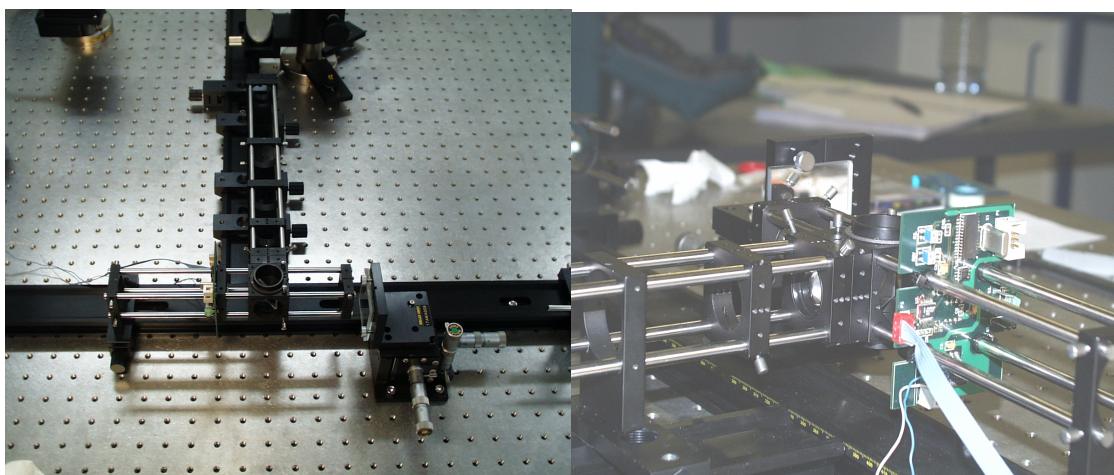


Fig. 5.2 – Montagem óptica com a placa PCB (esquerda) e o pormenor da sua integração no sistema (direita).

5.3. Avaliação da resposta axial do sistema

Neste estudo foram efectuados diversos testes em que diferentes comparações das configurações da montagem óptica foram realizadas. Todas as configurações no modo de reflexão epi-iluminada tendo-se usado como fonte de luz, em todas elas, um laser de diodo e como objecto o espelho plano. Os resultados foram analisados e visualizados numa aplicação desenvolvida em *Matlab* que permite uma melhor versatilidade no processamento dos dados.

5.3.1. Lente objectiva esférica versus acromática

Este primeiro teste foi realizado com a configuração do microscópio de campo amplo. Por isso dos componentes opcionais não foram utilizados os seguintes: *light shaping diffuser (LSD)*, fenda ajustável (*S*), *line generator (LG)* e o *pinhole (P)*. Foram então utilizadas, em alternativa, as duas lentes (asférica e acromática) como objectiva.

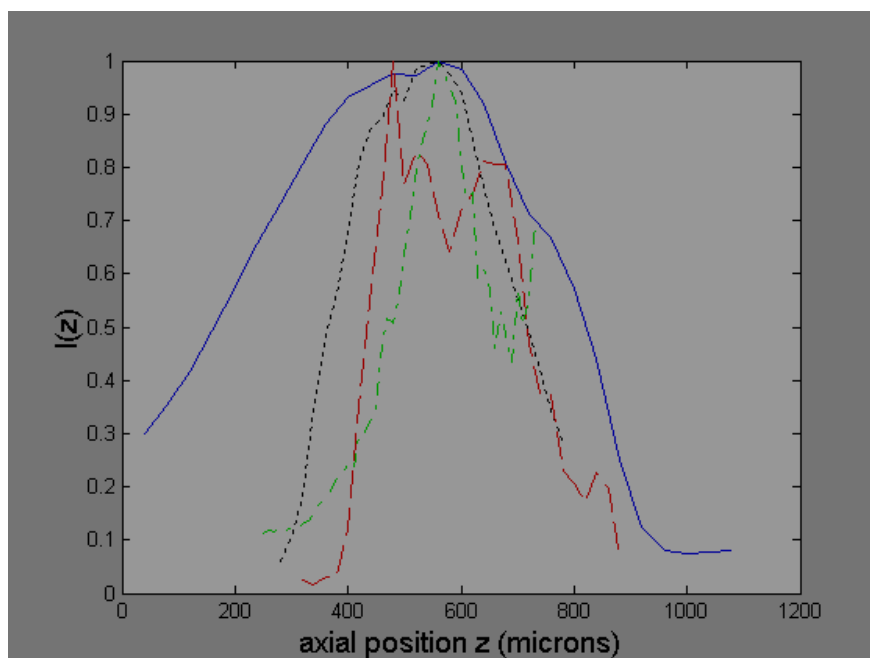


Fig. 5.3 - Resposta axial do plano reflector: objectiva acromática em campo amplo (linha azul), objectiva asférica em campo amplo (pontos pretos), com o detector pinhole de $100\mu\text{m}$ (traços vermelhos) e com iluminação em linha (traços verdes).

Podemos ver na figura 5.3 a distribuição da intensidade da luz em função da posição axial. Comparando os resultados da utilização das lentes esférica e acromática como lente objectiva, conclui-se que com a lente objectiva esférica consegue-se ter melhor resolução do que no caso da acromática já que se obtém uma distribuição de luz mais concentrada na parte central, isto é, o pico no gráfico é mais estreito. Portanto o *FWHM* é menor no caso da lente esférica e consequentemente tem melhor resolução axial.

Na figura 5.4 apresentam-se os gráficos 3D de visualização da distribuição da intensidade da luz no plano XZ. A direcção de X é a do sensor, logo corresponde aos pixels. A dos Z é a axial, logo é visualizada a intensidade de luz recolhida no sensor para diferentes posições axiais do objecto. Podemos ver que com esta configuração, em campo amplo e com filtro de polarização, quando se utiliza a lente esférica a distribuição da intensidade da luz é mais intensa e concentrada do que utilizando a lente acromática. Por esta razão optou-se por usar a lente esférica em detrimento da acromática como lente objectiva do sistema

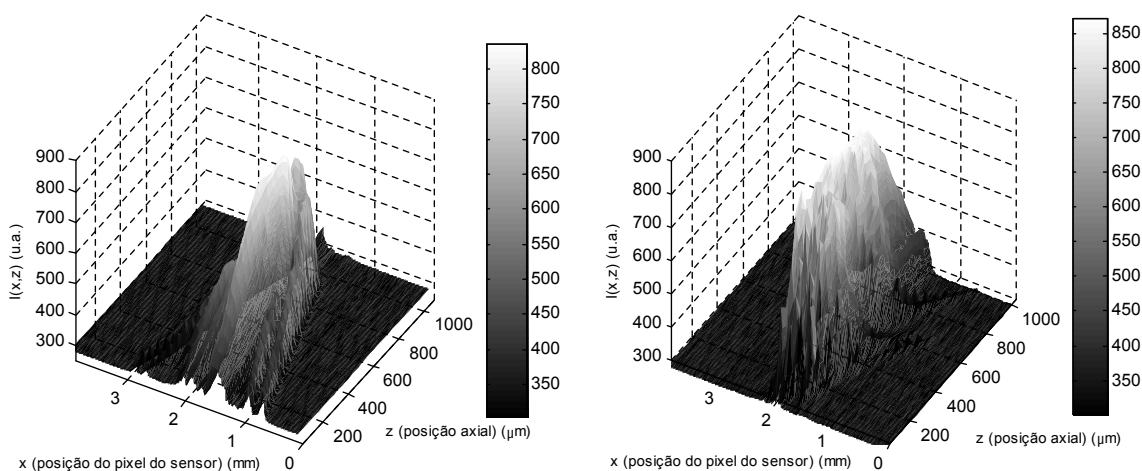


Fig. 5.4 - Distribuição de intensidades de luz no sensor com objectiva acromática (esquerda) e esférica (direita) em campo amplo (com filtros ND).

As distribuições da intensidade da luz com as duas objectivas, mais nítida para a lente acromática, têm um comportamento irregular o que pode ser, em larga medida, devido às aberrações (maioritariamente esféricas) das lentes que são consequência natural de termos usado componentes ópticos de baixo custo. Outras causas são a

imprecisão e instabilidade no alinhamento e também flutuações na potência com instabilidades pontuais da fonte de luz laser. Consequentemente as expressões bem conhecidas para a resolução lateral e axial de um sistema limitado pela difracção (*diffraction limited*) não reflectem a situação real do nosso sistema. Existe ainda um outro factor que é o de, devido à baixa qualidade das lentes usadas como objectiva, e para reduzir as aberrações esféricas se ter tido o cuidado de o feixe não preencher a abertura da objectiva, e por isso a sua *NA* efectiva é consideravelmente menor que o valor nominal apresentado.

5.3.2. Iluminação em linha versus *pinhole* de detecção

Neste segundo teste foram usadas as configurações ópticas com detector *pinhole* (*P*) circular de 100 μ m em frente do sensor linear e com iluminação em linha. Utilizou-se apenas a lente asférica como objectiva.

Refira-se que para obter a iluminação em linha desejada, isto é, a formação do feixe planar foram testados os seguintes componentes: lente anamórfica de baixo custo, uma fenda ajustável, e um *LSD* 0,2°x40°. Para a comparação seguinte considerou-se a configuração em que foi utilizada a lente anamórfica (*LG* na figura 5.1).

A comparação da resposta axial, usando o plano reflector, em termos de distribuição de intensidades no eixo *z* (axial) entre a iluminação em linha e com um *pinhole* de 100 μ m mostra que com a iluminação em linha, como se vê no gráfico da figura 5.3 apresentada acima, surge um pico lateral para posições de reflexão, no eixo dos *z*, de maior profundidade enquanto que com o *pinhole* aparece o efeito de interferência conhecido por *speckle*. O *speckle* é um padrão de intensidade aleatória produzido por interferências mútuas (sobreposição de duas ou mais ondas) de ondas coerentes com diferentes fases ou sujeitas a flutuações de intensidades.

A aparência do *speckle* clássico é na sua maioria uma parte negra com ilhas de luz. A distribuição do *speckle* é dependente do ângulo de observação. Grandes ângulos dão uma função de autocorrelação pequena e um ângulo pequeno dá um grande caminho para uma grande distância de visão.

5.3.3. Pinhole de detecção: 10 μm vs 100 μm

Na figura 5.5 podemos ver o resultado da utilização de um *pinhole* com diferentes aberturas (10 μm e 100 μm) bem como uma comparação entre as duas utilizações. A utilização de um *pinhole* faz com que se tenha maior precisão mas aparecem mais interferências o mesmo se verifica em menor escala num circuito em campo amplo.

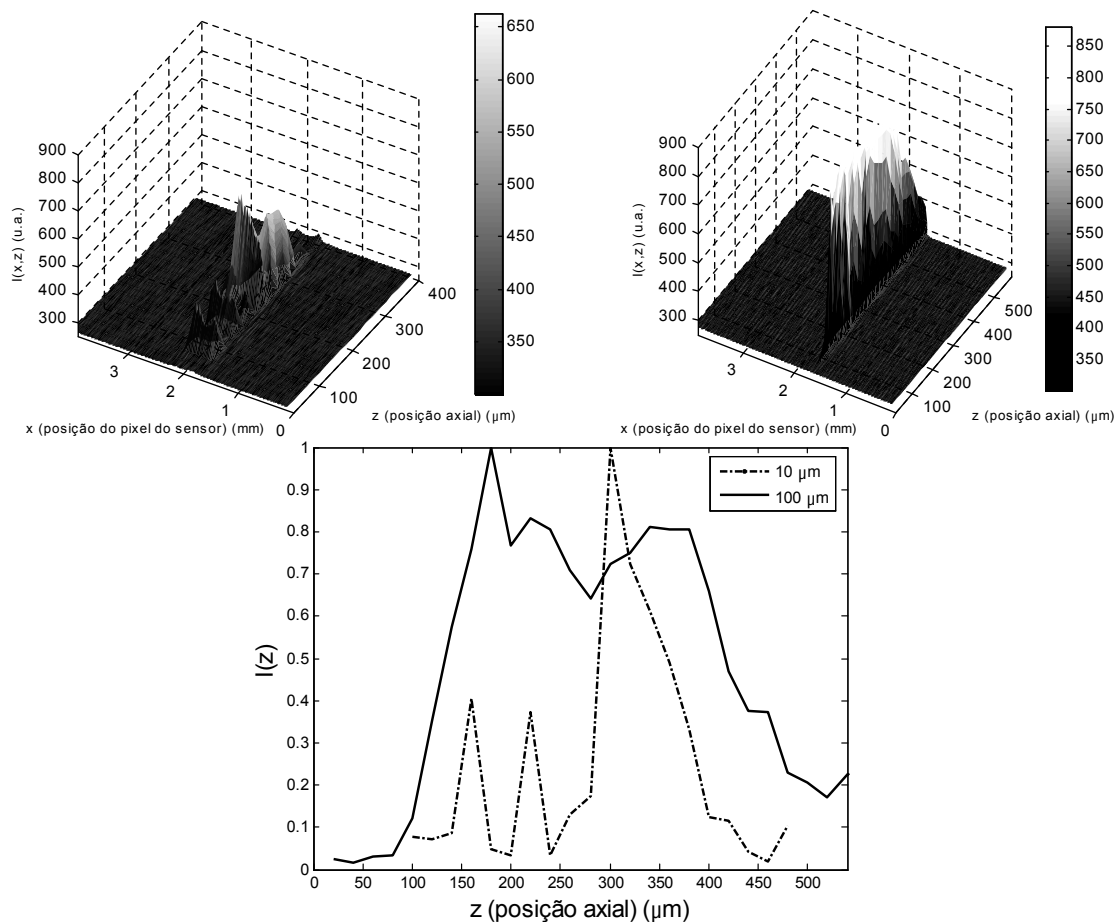


Fig. 5.5 - Distribuição de intensidades de luz no sensor com objectiva esférica com um *pinhole* de 10 μm (esquerda) e um *pinhole* de 100 μm (direita) e em baixo a resposta axial do plano reflector que compara a utilização dos dois *pinholes* de 10 μm e 100 μm .

Pode-se concluir que a utilização de um *pinhole* de 10 μm tem a vantagem, em relação á utilização de um *pinhole* de 100 μm , de o feixe de luz recolhido sensor não estar tão disperso. Contudo para se determinar a verdadeira resolução confocal o *pinhole* de detecção deve ser infinitamente pequeno o que resulta num sinal pequeno, como já se percebe pela figura 5.5 para o caso do *pinhole* de 10 μm em comparação com

o de $100\mu\text{m}$. Ao usar-se um *pinhole* com uma abertura maior para se obter um sinal melhor degrada-se a resolução do sistema. Por isso um compromisso entre a resolução e a razão sinal-ruído deve ser feita em cada aplicação específica. Por outro lado, apesar de com a utilização de um *pinhole* de $10\mu\text{m}$ se obter maior precisão do que com um de $100\mu\text{m}$, não é aconselhável utilizar um *pinhole* mais estreito uma vez que surgiriam mais interferências (*speckle*). O efeito de *speckle* já é visível na figura 5.5, no gráfico de comparação do uso dos dois *pinholes* de diferentes larguras.

5.3.4. Efeito da utilização de um *LSD*

O facto de se usar uma fonte de luz laser, conseqüentemente luz com grande comprimento de coerência, induz interferência por isso foi também testada a aplicação do *LSD* que resultou na eliminação deste tipo de *speckle* como se pode ver na figura 5.6. Comparando as distribuições de intensidade de luz no sensor (com objectiva esférica em campo amplo), usando *LSD* ou sem *LSD*, verifica-se que os picos de interferência que aparecem no caso em que não se usa o *LSD* desaparecem quando se introduz o *LSD* apesar de ser à custa de um alargamento no pico (aumento da *FWHM*). Pode-se concluir que o *speckle* é assim eliminado, melhorando a resposta do sistema óptico, na medida em que se consegue obter uma distribuição da luz mais uniforme.

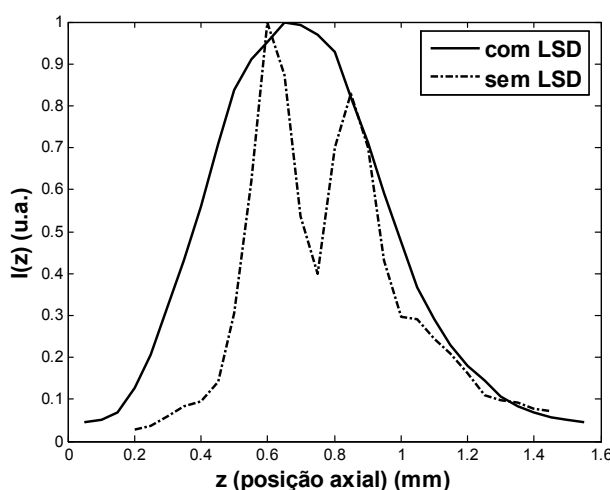


Fig. 5.6 - Distribuição de intensidades de luz no sensor com objectiva esférica em campo amplo usando *LSD* ou sem *LSD*.

5.3.5. Filtros de polarização vs filtros *ND*

Na figura 5.7 temos a comparação da distribuição de intensidades entre o uso de filtros de polarização e *ND*. Usou-se a objectiva esférica em campo amplo com *LSD*. Não se verificou grande alteração de uma configuração para a outra, isto é, obtiveram-se resultados semelhantes com os dois tipos de filtros.

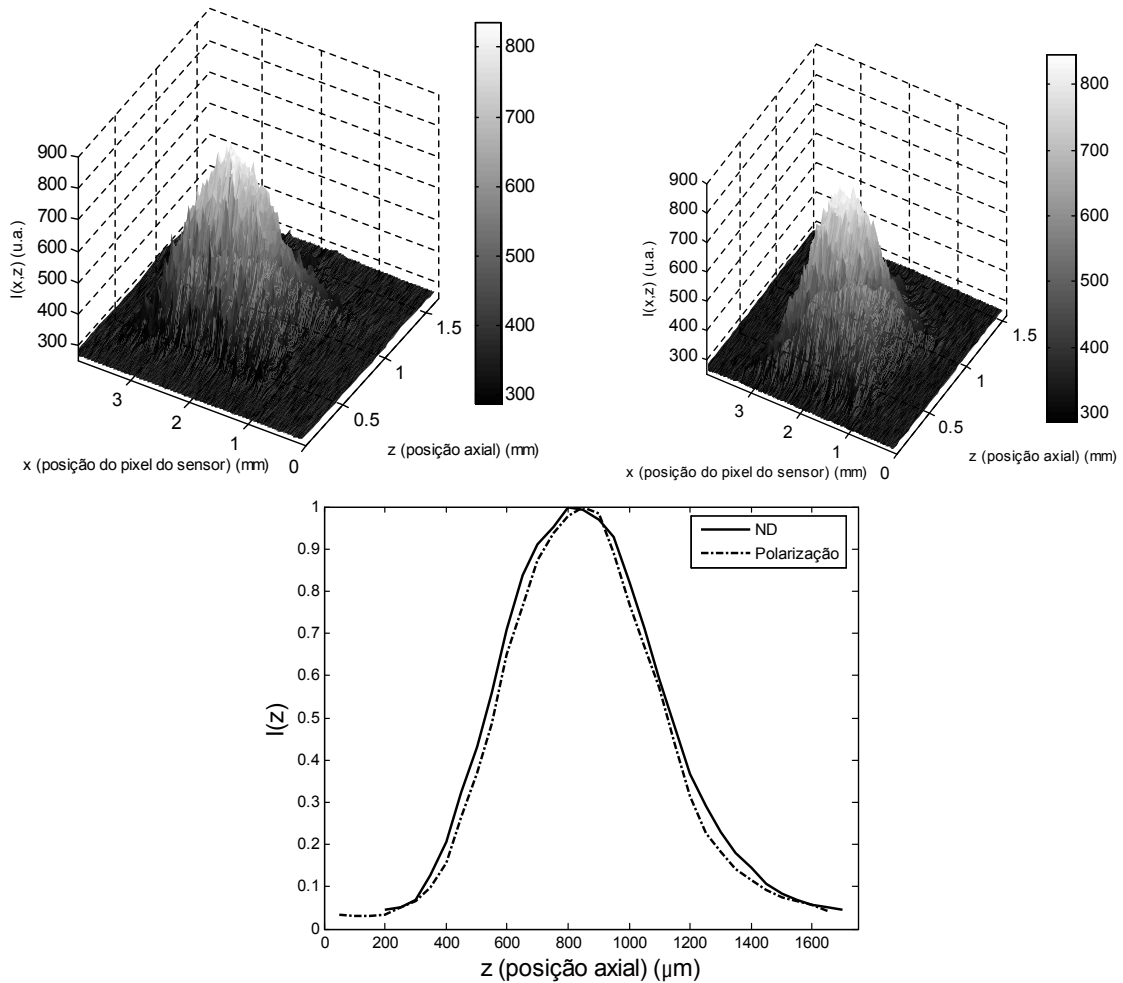


Fig. 5.7 - Distribuição de intensidades de luz no sensor com objectiva esférica em campo amplo com *LSD* e *ND* (esquerda) e com *LSD* com filtro de polarização (direita) e em baixo a resposta axial do plano reflector que compara as duas configurações.

5.3.6. Fenda com 530 μm vs 540 μm

Em vez de um detector circular (*pinhole*) usou-se um detector em fenda que preserva a capacidade confocal nos eixos perpendiculares á fenda e tem um

comportamento convencional na direcção paralela o que resulta uma função *PSF* assimétrica. A fenda melhora a resolução espacial em particular no eixo dos *Z* (axial) e a distribuição de intensidades da luz. Na figura 5.8 podemos ver a intensidade de luz no sensor com uma fenda de $540\mu\text{m}$ e $530\mu\text{m}$. A fenda de $530\mu\text{m}$ apresenta uma intensidade de luz inferior ao verificado com a de $540\mu\text{m}$, no entanto com esta fenda de $530\mu\text{m}$ consegue-se uma melhor resolução axial como se vê na figura 5.8. Podemos concluir que em termos de resolução axial a fenda com $530\mu\text{m}$ é ligeiramente superior á fenda de $540\mu\text{m}$ apesar desta apresentar maior intensidade de luz. Tem portanto melhor relação sinal-ruído mas se a fenda for muito maior pode provocar uma saturação do sensor.

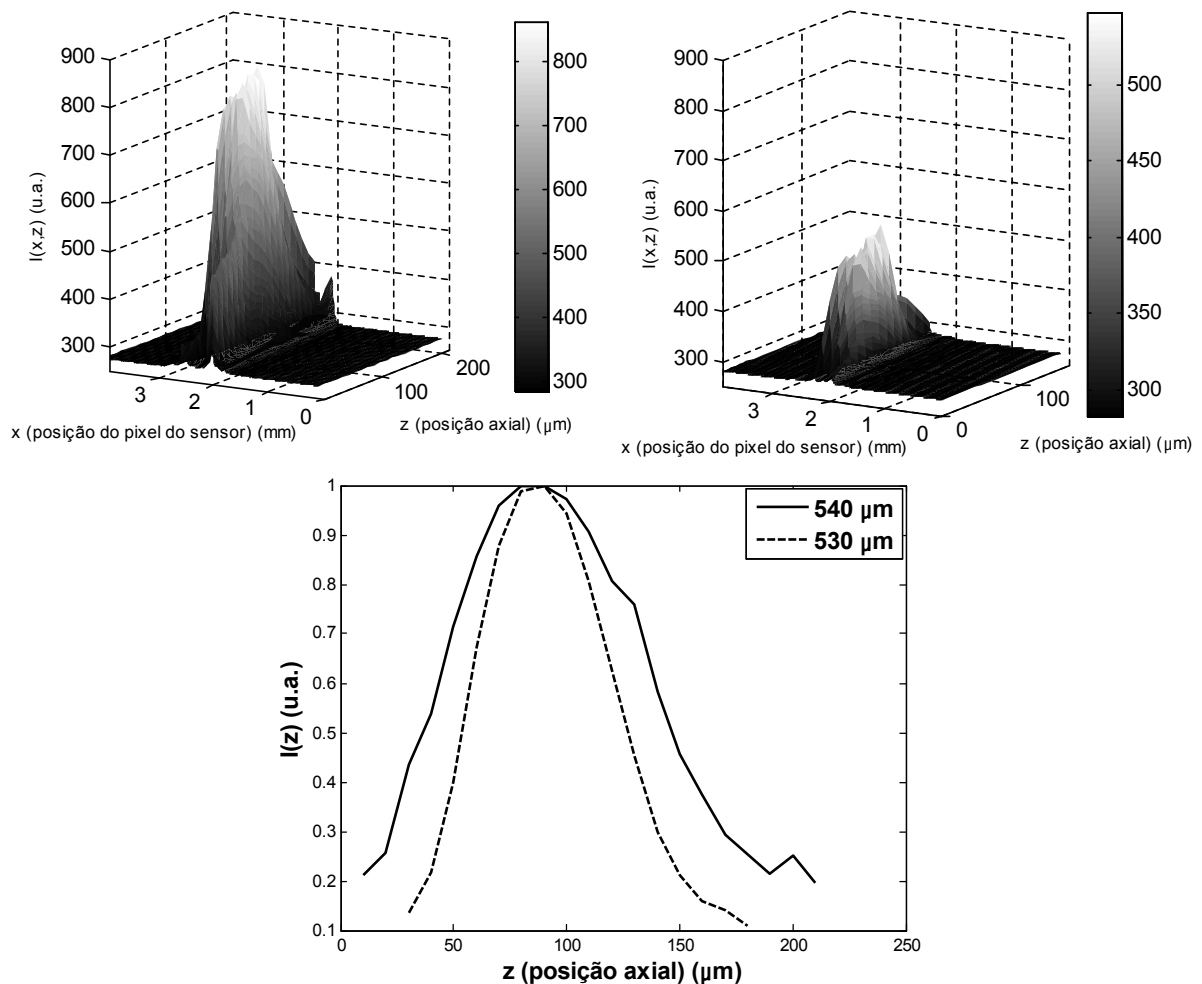


Fig. 5.8 - Distribuição de intensidades de luz no sensor com objectiva esférica com uma fenda ajustável com abertura de $540\mu\text{m}$ (esquerda) e uma fenda de abertura de $530\mu\text{m}$ (direita) e em baixo a resposta axial do plano reflector que compara a utilização as duas fendas de abertura de $540\mu\text{m}$ e $530\mu\text{m}$

6. Conclusões e Trabalho Futuro

6.1. Conclusões

O objectivo do projecto global em que este trabalho se insere é desenvolver uma técnica de baixo custo para um microscópio para obtenção de imagens *3D* com um tempo de aquisição baixo e uma razão sinal ruído alta em relação ao microscópio confocal convencional. É usado para tal um sensor *CMOS* linear e será desenvolvido um algoritmo capaz de extrair informação *3D* da distribuição de intensidade da luz colectada.

De entre os microscópios de obtenção de imagem *3D* o microscópio confocal é conhecido pelo seu desempenho na resolução axial, isto é, a sua capacidade de realizar o seccionamento óptico. Num microscópio convencional a restrição em profundidade é limitada pela profundidade de foco do instrumento. Objectos fora do plano focal aparecem esbatidos mas contribuem para a imagem final.

A maior vantagem do microscópio confocal é a capacidade de discriminação em profundidade que é estabelecida principalmente pelo tamanho do *pinhole* confocal, mas dependendo também das características da objectiva, e em particular da sua abertura numérica, para eliminar as aberrações esféricas. Usa-se normalmente uma fonte de luz laser que é focada na amostra após atravessar um *pinhole*. O seu maior inconveniente é o elevado tempo de aquisição que decorre da necessidade de efectuar o varrimento (*scanning*) dos três eixos e o fraco sinal detectado devido á rejeição por parte do *pinhole* de grande parte da energia da luz incidente. Uma das formas de minimizar este ponto fraco é mudar da obtenção de imagem feita através do varrimento ponto a ponto para um sistema com varrimento por fenda. Sabe-se que a fenda tem um efeito na discriminação da profundidade focal entre o microscópio convencional e o confocal^[59,60].

Esta foi a principal motivação para o projecto em que este trabalho se inclui, em que se usa um sensor *CMOS* linear funcionando, por isso, como uma fenda logo a

obtenção de imagem dos detalhes que correm paralelamente ao sensor é semelhante à do microscópio convencional enquanto que para os que são perpendiculares à fenda (sensor) se assemelha à obtenção de imagem confocal. Neste *SOM* a necessidade de varrimento mecânico na direcção paralela ao sensor linear é eliminada e tenta-se melhorar a sua resolução axial através da aplicação de técnicas de processamento de sinal, com recurso à informação colectada nos *pixels* do sensor linear e à variação da distribuição da intensidade com um varrimento axial. Pretendeu-se assim obter um sistema com características próximas da do confocal sem utilizar *pinholes*.

O sensor linear foi então usado como detector deste protótipo laboratorial num arranjo epi-iluminado no modo de reflexão num tipo de varrimento da plataforma (objecto) com o propósito de colectar a luz laser reflectida num objecto planar perfeito. Este objecto é movimentado através do eixo óptico, aproximando-se e afastando-se do plano focal do *layout* confocal. A opção pelo tipo de varrimento da plataforma foi então tomada com o objectivo de diminuir os efeitos de aberrações, pois assim a fonte de luz é estacionária passando a luz sempre na parte mais central das lentes. Mesmo assim, sendo as objectivas usadas de baixo custo, são incapazes de produzir imagens com resolução próxima do seu valor teórico pois de modo a reduzir a aberração esférica é necessário usá-las na parte mais central, levando à diminuição da abertura numérica efectiva.

Neste trabalho foram testadas várias configurações para a montagem óptica em que, como resultado final, foi encontrada a configuração que melhor se enquadra no nosso estudo, e que se pode ver na figura 6.1.

Comparando os valores aproximados da *FWHM* das curvas da intensidade em função da posição axial, $I(z)$, para cada configuração podem-se reter os seguintes aspectos:

- Neste caso em que se usou uma fonte de luz laser, a lente esférica tem vantagem sobre a acromática pois a *FWHM* tem valores de 400 μm e 600 μm , respectivamente.
- Com a configuração do campo amplo a *FWHM* tem o seu valor máximo (aprox. 600 μm) o que é elevado tendo-se obtido com iluminação em linha (lente anamórfica) uma *FWHM* de 300 μm .

- A utilização de um *pinhole* de 10 μm permite obter um *FWHM* melhor do que com o de 100 μm (valores de 100 μm e 300 μm , respectivamente) mas a relação *S/N* degrada-se para cerca de metade
- Da utilização da fenda resulta a melhor *FWHM* de todas as configurações se a sua largura for de 530 μm (ligeiramente inferior a 100 μm) enquanto que com a largura de 540 μm fica-se por um valor cerca de 50 μm superior. Contudo isto é conseguido à custa da redução da relação *S/N* para aproximadamente 1/3.
- A utilização de um *LSD* consegue ter o efeito de eliminar o *speckle* obtendo-se, no entanto *FWHM* maiores (aproximadamente de 500 μm).

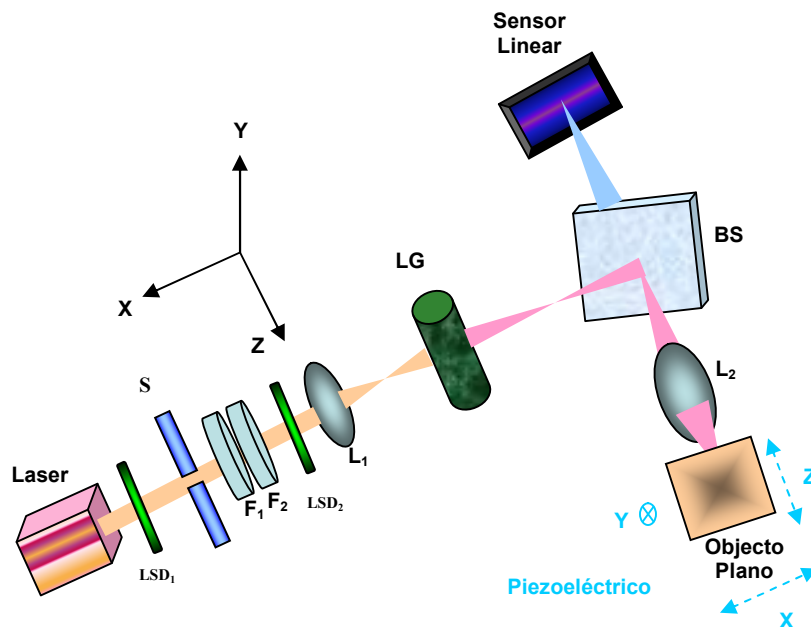


Fig. 6.1 – Esquema final da montagem óptica

As distribuições de intensidade de luz recolhidas no sensor *CMOS* do tipo matriz linear variam, alargando e estreitando, à medida que o plano objecto se afasta ou aproxima do plano focal, que é uma consequência da equação dos focos conjugados.

Por outro lado foi visível que o grande comprimento de coerência da luz laser induz interferência, normalmente designada por *speckle*, mais nítida quando se usam os *pinholes*. Por isso, mais testes foram realizados para avaliar a utilização do *LSD* para eliminação deste tipo de *speckle*, tendo os resultados confirmado a sua eficácia. Como resultado da utilização de uma fonte de luz incoerente branca (como uma lâmpada de

halogéneo) ter-se-ia a eliminação deste *speckle* contudo a resolução axial ou a capacidade do seccionamento óptico do sistema seria danificado pela aberração cromática longitudinal que inevitavelmente surgiria.

Sob o ponto de vista mais prático o sistema permitiu obter resultados consistentes, isto é, verificou-se que não era necessário efectuar o alinhamento óptico sempre que se pretendia usar o sistema seleccionando a melhor combinação dos componentes ópticos. Os resultados não eram alterados de forma significativa quando se repetiam após se terem efectuado várias alterações na configuração da montagem óptica.

Foi possível identificar algumas tendências nas variações de distribuição de luz com a posição axial que permitem esboçar o comportamento de uma frente de onda. Apesar de não se ter obtido um resultado quantitativo, em particular da resolução axial, parece claro que outro método diferente de definir esse parâmetro baseado noutras medidas ou mesmo procedimentos deve ser seguida.

6.2. Trabalho futuro

O trabalho futuro irá envolver o desenvolvimento de uma nova plataforma de *hardware*, de aquisição e controlo, totalmente integrada e supervisionada por um *PC*. Irá atingir uma taxa de aquisição de 1000 *frames/s* com 1024 *pixels* (4000 ou 8000 *frames/s* para 256 e 128 *pixels* respectivamente). Este novo sistema será capaz de controlar e sincronizar diferentes componentes na montagem óptica particularmente o sensor linear e a fenda, por isso poderá realizar um varrimento completo, nos eixos X e Y , em linha em vez de o fazer com um ponto de iluminação. A ideia é obter uma plataforma universal que torne possível a interface com diferentes sensores do tipo linear ou de área, bem como posicionadores tornando possível a anteriormente mencionada aplicação de um *SLM*.

Tanto o *hardware* como o *software* para configurar o modo de funcionamento do sensor, controlar parâmetros de aquisição, posicionar a plataforma do objecto com piezoeléctrico e (ou) comandar outros mecanismos de varrimento devem ser incluídos

no sistema de modo a obter uma maior automatização e completa integração das suas funcionalidades.

Para que a velocidade de comunicação entre o *PC* e o sistema seja maximizada poderá ser usada uma comunicação por *USB* ou o desenvolvimento de uma nova placa *ISA/PCI* em que a taxa de transferência de dados é maior.

Em paralelo com o trabalho experimental uma pesquisa intensiva tem que ser continuada focando-se nas técnicas de processamento de sinal (em particular na desconvolução) com vista a actualizar e testar permanentemente a aplicação *Matlab* a ser desenvolvida para o efeito. Esta aplicação deverá usar a informação óptica colectada num sensor linear, enquadrada com as expressões da óptica de *Fourier*, e especialmente dedicado às propriedades da formação da imagem de sistemas com iluminação em linha. A utilização de um segundo sensor para se obter informação do ponto de iluminação é uma ideia para melhorar o resultado das técnicas de desconvolução.

Além disso é possível apontar algumas direcções para melhorar o sistema óptico para que possamos concretizar o objectivo de demonstrar que se pode ser bem sucedido através da exploração de técnicas de processamento de sinal aplicadas aos dados recolhidos num sensor linear. Desde logo através do uso de uma lente objectiva com características de desempenho adequadas para a obtenção de imagem em modo de reflexão. Também a aplicação de algum tipo de modulação na luz de iluminação como a utilização de uma fonte de luz em linha (*line generator*) que assegure uma boa qualidade na forma e tamanho da mancha de iluminação. Por fim a utilização de luz branca com vista a eliminar a ocorrência de franjas de interferência, que são características de um microscópio no modo de reflexão quando se usa uma fonte de luz laser.

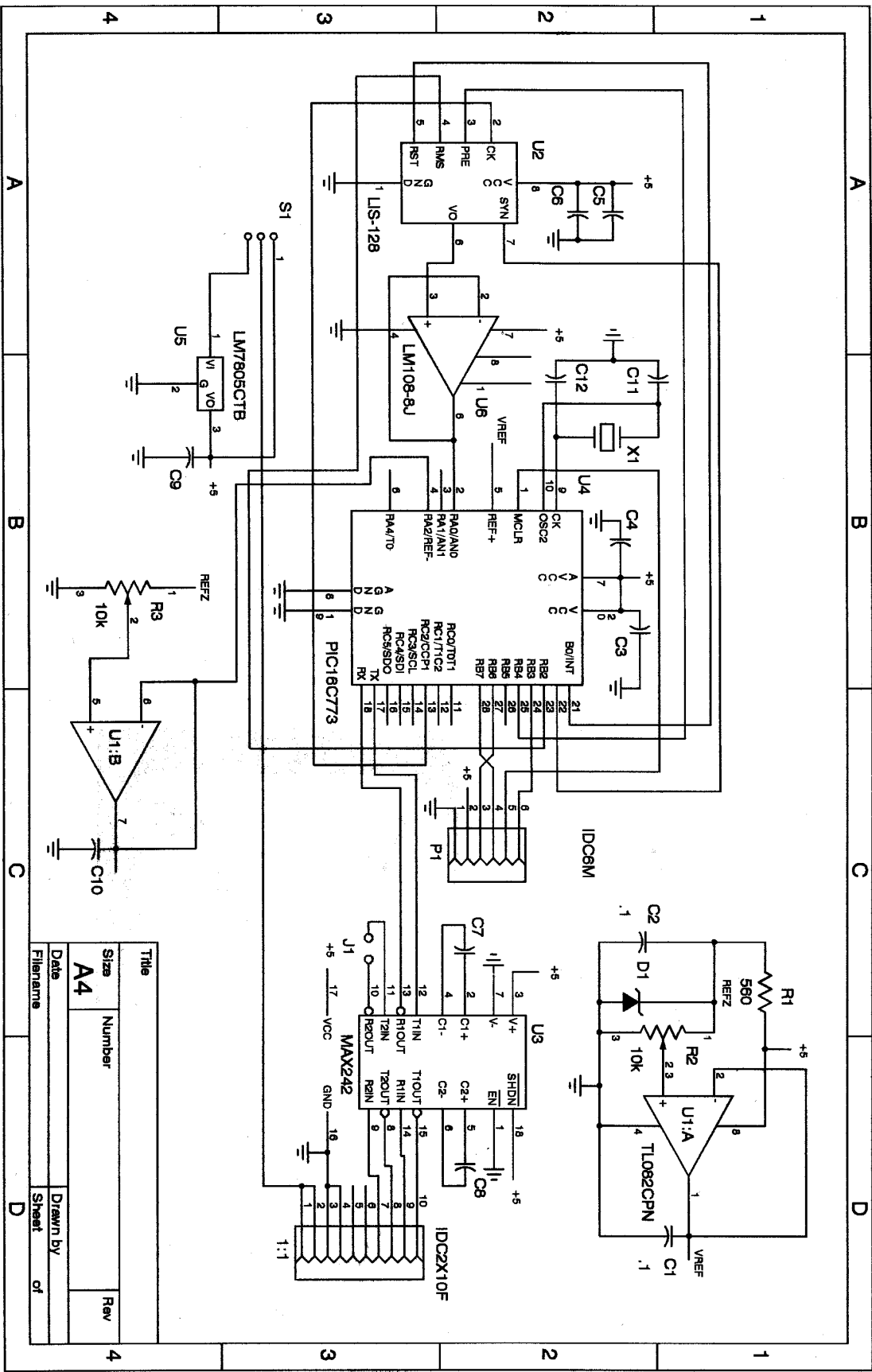
-
- [1] – J. Paul Robinson, *Introduction to Confocal Microscopy and Image Analysis*, Purdue University (1993).
- [2] – J. B. Pawley, *Handbook of Biological Confocal Microscopy*, 2nd edition, Plenum Press - New York (1995).
- [3] – Joseph L. Bruneni, *An Illustrated History of the American Ophthalmic Industry*, Optical Laboratories Association (1994).
- [4] – R. Clay, T. Court, *The History of the Microscope*, Holland Press (1975).
- [5] – W. G. Hartley, *The Light Microscope: Its Use and Development*, Senecio Publishing Co. (1983).
- [6] – Eugene Hecht, *Óptica*, Fundação Calouste Gulbenkian (1991).
- [7] – Jay Reichman, *Handbook of Optical Filters for Fluorescence Microscopy*, Chroma Technology Corp (2000).
- [8] - Marvin Minsky, *Memoir on Inventing the Confocal Scanning Microscope*, Published in *Scanning*, vol.10 pp128-138, (1988).
- [9] – M. Abramowitz, *Fluorescence Microscopy: The Essentials*, Olympus America Inc. (1993).
- [10] – S. Inoué, *Video Microscopy*, Plenum Press – New York (1986).
- [11] –J. Lakowicz, *Principles of Fluorescence Spectroscopy*, Plenum Press – New York (1983)
- [12] – T. Wilson & C. Sheppard, *Teory and Practice of Scanning Optical Microscopy*, Academic Press - London (1984).
- [13] – J. W. Lichtman , *Confocal Microscopy*, *Scientific American* 271, 40-45 (1994).
- [14] - T. Wilson, *Confocal Microscopy*, Academic Press (1990).
- [15] - M. Born and E. Wolf, *Principles of Optics*, Pergamon Press Oxford 6th ed. (1986).

-
- [16] – H. H. Hopkins, *On the diffraction theory of optical images*, Proc. Roy. Soc. Lond., A 217, 408-432 (1953).
- [17] - W. Lukosz, *Optical systems with resolving powers exceeding the classical limit*, J. Opt. Soc. Am., 56, 1463-1472 (1966).
- [18] – E. A. Ash and G. Nicholls, *Super resolution aperture scanning microscopy*, Nature, 237, 510-512 (1972).
- [19] – G. W. McCutchen, *Super resolution microscopy and the Abbé resolution limit*, J. Opt. Soc. Am., 57, 1190-1192 (1967).
- [20] - Marvin Minsky, *Microscopy Apparatus*, United States Patent 3,013,647, Dec. 19, 1961, (Filed Nov. 7, 1957).
- [21] – S. C. Baer, *Optical Apparatus Providing Focal Plane Specific Illumination*, United States Patent 3,547,512, (1970).
- [22] – C. J. R. Sheppard and T. Wilson, *Effects of High Angles of Convergence on $V(z)$ in the Scanning Acoustic Microscope*, Appl. Phys. Letters, 38, 858-860 (1981).
- [23] – T. Wilson and D. K. Hamilton, *Dynamic Focussing in the Confocal Scanning Microscope*, J. Microscope, 128, 139-143 (1982).
- [24] – H. J. Matthews, *D. Phil. Thesis*, University of Oxford (1987).
- [25] – van der Voort and All, *Three Dimensional Visualisation Methods for Confocal Microscopy*, J. Microscope, 153, 123-132 (1989).
- [26] – D. K. Hamilton and T. Wilson, *Optical Sectioning in Infra-Red Scanning Microscopes*, Proc. I.E.E., 134 Pt. I, 85-86 (1987).
- [27] – T. Wilson and C. J. R. Sheppard, *Theory and Practice of Scanning Optical Microscopy*, Academic Press London (1984).
- [28] – J. W. Goodman, *Introduction to Fourier Optics*, second edition, McGraw Hill Companies (1998).
- [29] – Robert H. Webb, *Confocal Optical Microscopy*, Rep. Prog. Phys., 59, 427-471 (1996).

-
- [30] – Mário Ferreira, *Óptica e Fotónica*, Lidel – Edições Técnicas (2003).
- [31] – A. J. de Dekker and A. van den Bos, *Resolution: a survey*, Op. Soc. Of America., Vol. 14, N°3/March, 547-557 (1997).
- [32] – Corral, M. Martinez et al., *Effective Fresnel-Number Concept for Evaluating the Relative Focal Shift in Focused Beams*, J. Op. Soc. Am. A., Vol. 15, N°2/February, (1998).
- [33] – A. A. Michelson, *Studies in Optics*, University Chicago Press (1927).
- [34] – R. H. Webb and C. H. Dorey, *Pixelated Image*, New York Plenum pp 41-51 (1990).
- [35] – D. R. Sandisson and W. W. Webb, *Background Rejection and S/N Optimization in Confocal and Alternative Fluorescence Microscopy*, App. Opt. 33, 603-615 (1994).
- [36] – P. J. Verveer et al., *Theory of confocal fluorescence imaging in the programmable array microscope (PAM)*, J. of Microscopy, 189, 192-198 (1997).
- [37] – D. T. Dewer et al., *Direct view microscopy: experimental investigation of the dependence of the optical sectioning characteristics on pinhole array configuration*, J. of Microscopy, 187, 54-61 (1997).
- [38] – M. E. Motamedi et al., *Micro-optoelectro-machanical devices on-chip optical processing*, Opt. Eng., 36, 1282-1297 (1997).
- [39] – D. A. Agard, *Optical sectioning microscopy: cellular architecture in three dimensions*, Annu. Rev. Biophys. Bioeng., 13, 191-219 (1984).
- [40] – P. A. Benedetti et al., *Confocal line microscopy*, J. of Microscopy, 170, 119-129 (1992).
- [41] – T. Wilson and S. J. Hewlett, *Imaging in scanning microscopes with slit-shaped detectors*, J. of Microscopy, 160, 119-129 (1990).

- [42] – J. Mitic et al., *Optical sectioning in wide-field microscopy obtained by dynamic structured light illumination and detection based on a smart pixel detector array*, Optics Letters, Vol. 28, nº9, 698-700 (2003).
- [43] – Mc. Nally et al., *Artefacts in computational optical-sectioning microscopy*, J. Opt. Soc. Am. A, Vol. 11, 1056-1067 (1994).
- [44] – M. A. A. Neil et al., *Method of optical sectioning by using structured light in a conventional microscope*, Opt. Letters, Vol. 15 N°24, 1905-1907 (1997).
- [45] – C. G. Morgan et al., *Optical sensor for imaging an object*, U. S. Patent 5,381,236 (1995).
- [46] – G. Q. Xiao et al., *Real-time confocal scanning microscope*, Appl. Phys. Lett. 53(8), 716-718 (1988).
- [47] – C. J. Cogswell et al., *Three dimensional microscopy: image acquisition and processing*, Proc. SPIE, Vol. 2655, 198-208 (1996).
- [48] – You Zheng et al., *Novel two-dimensional position measurement method with linear charge couple device*, Optical Eng., Vol. 37, N°9, 2601-2604 (1998).
- [49] – W. Cai, Y. Li and Z. Xu, *CCD Technology and Application*, Electronic Industry Publication, Beijing (1990).
- [50] – J. J. Lixia, *A method of detecting position of images on CCD*, Optical Technol., N°5, 21-27 (1996).
- [51] – Thorsten Graeve and Gene P. Weckler, *High-resolution CMOS imaging detector*, Physics of Medical Imaging, SPIE Vol. 4320, (2001).
- [52] – Youngjoong Joo et al., *Smart CMOS Focal Plane Arrays: A Si CMOS Detector Array and Sigma-Delta Analog-to-Digital Converter Imaging System*, IEEE Journal, Vol. 5, N°2 (1999).
- [53] – Dave Litwiller, *CCD vs. CMOS: Facts and Fiction*, Photonics Spectra – Laurin Publishing Co. (2001).
- [54] – <http://www.olympusfluoview.com/java/index.html>

-
- [55] – C. J. R. Sheppard and D. M. Shotton, *Confocal Laser Scanning Microscopy*, BIOS Scientific Publishers, Oxford (1997).
- [56] – P. J. Shaw, *Comparison of wide-field/deconvolution and confocal microscopy for 3D imaging*, BIOS Scientific Publishers, Oxford (1997).
- [57] - Milton P. Macedo, António J. Barata et al., *Development of a low-cost SOM laboratory prototype with line-illumination: evaluation of deconvolution techniques on its axial resolution*, Proceedings of SPIE-OSA Biomedical Optics, SPIE Vol. 5139, 245-252 (2003).
- [58] – Milton P. Macedo, António J. Barata et al., *Development and test of instrumentation using a linear sensor for confocal microscopy*, Focus on Microscopy 2003, 15th International Conference on Confocal Microscopy and 16th Int. Conf. on 3D Processing, Génova, Itália, Program and Abstract Book, pp 87, (2003).
- [59] – T. Wilson and S. J. Hewlett, *Imaging in scanning microscopes with slit-shaped detectors*, Journal of Microscopy, 160 (2), 115-139 (1990).
- [60] - Milton P. Macedo, António J. Barata, Ana G. Fernandes, Carlos M. Correia, *A wide-field microscopy technique using a linear image sensor for obtaining 3D images*, BiOS2005, SPIE International Symposium on Biomedical Optics, San Jose, USA, 22-27 January 2005, Proceedings of SPIE, Vol. 5701, 172-181, 2005.



Title		Number		Rev	
Size A4					
Date	Drawn by			Sheet	
Filarmite		of		4	


```

/*****
/*      File : confocal.h                                     */
/*      Description : Inicialização do Micro e configuração das comunicações      */
/*      Description file: Inicialização                                           */
/*      Compiler used : Borland C                                               */
/*      Written by: Antonio J. Barata                                           */
/*      Last revision: 07/2005                                                  */
/*      Update & revision file: 07/2005                                         */
*****/

/***** Definições genericas *****/
#ifndef TRUE
#define TRUE 1
#endif
#ifndef FALSE
#define FALSE 0
#endif
// Codigos devolvidos por getkey
#define RET    0x0D    // Return

/***** UG-AUXIL.C *****/
void termina(void);
void fatalerror(char *m1,char *m2);
void soundwarning(int sd);

/***** Variavel global *****/
extern char *comerrmsg[];    // Mensagens de erro

/***** Gestao do ficheiro de configuracao *****/
#define MAXPRMLEN 200    // tamanho maximo de um parametro
// informacao de configuracao das comunicacoes
typedef struct {
    unsigned char serialport;
    unsigned char irqnumber;
    unsigned char baudrate;
    unsigned char databits;
    unsigned char stopbits;
    unsigned char parity;
    unsigned char config;
}comcfginfo;
extern comcfginfo ccfg;
int rdcfgprm(char *cfgfile,long *fpos,char *prmname,char *prm);
int rdcomcfg(char *cfgfile,comcfginfo *ccpt);

/*****
/*      Alteracao dos tipos predefinidos                                     */
*****/
#ifndef _WIN32
#define ur_char    __int8
#define ur_byte    unsigned __int8
#define ur_int     __int16
#define ur_word    unsigned __int16
#define ur_long    __int32
#define ur_dword   unsigned __int32
#define ur_float   float
#define ur_double  double
#else
#define ur_char    char

```

```
#define ur_byte unsigned char
#define ur_int int
#define ur_word unsigned int
#define ur_long long
#define ur_dword unsigned long
#define ur_float float
#define ur_double double
#endif

/***** Configuracao das comunicacoes *****/
#define COM1 0
#define COM2 1
#define COM3 2
#define COM4 3
#define IRQN 0
#define IRQ3 1
#define IRQ4 2
#define BAUD110 0X00
#define BAUD150 0X20
#define BAUD300 0X40
#define BAUD600 0X60
#define BAUD1200 0X80
#define BAUD2400 0XA0
#define BAUD4800 0XC0
#define BAUD9600 0X0C
#define BAUD19200 0X06
#define BAUD38400 0X03
#define BAUD57600 0X02
#define BAUD115200 0X01
#define PARYNONE 0X00
#define PARYODD 0X08
#define PARYEVEN 0X18
#define STOP1BIT 0X00
#define STOP2BIT 0X04
#define DATA7BIT 0X02
#define DATA8BIT 0X03
#define NOCONFIG 0X00 // erro no ficheiro de configuracao
#define LOCONN 0X01 // ligacao local
#define REMCONN 0X02 // ligacao remota
#define MDMCONN 0X03 // ligacao via modem

/***** Programacao das comunicacoes *****/
#define COMPORT1 0X03F8 // Endereco base de COM1
#define COMPORT2 0X02F8 // Endereco base de COM2
#define COMPORT3 0X03E8 // Endereco base de COM3
#define COMPORT4 0X02E8 // Endereco base de COM4
#define COMPORT1MEN 0XEF // Comunicacoes autorizadas - COM1 no 8259
#define COMPORT2MEN 0XF7 // Comunicacoes autorizadas - COM2 no 8259
#define COMPORT1INT 0X0C // Vector de interrupcao do COM1
#define COMPORT2INT 0X0B // Vector de interrupcao do COM2
#define SDLAB 0X80 // Mascara para DLAB = 1
#define RDLAB 0X7F // Mascara para DLAB = 0
#define ENRXRDY 0X01 // Mascara para permitir RX READY interrupt
#define ENTXRDY 0X02 // Mascara para permitir TX READY interrupt
#define ENOUT2 0X08 // Mascara para permitir interrupcoes - 8250
#define SRTS 0X02 // Mascara para activar RTS
#define SDTR 0X01 // Mascara para activar DTR
```

```

#define DR    0    // Registo de dados
#define IER    1    // Interrupt Enable Register
#define BAUD0  0    // Low byte - DLAB = 1 em LCR
#define BAUD1  1    // High byte - DLAB = 1 em LCR
#define IIR    2    // Interrupt identifier register
#define LCR    3    // Line control register
#define MCR    4    // Modem control register
#define LSR    5    // Line status register
#define MSR    6    // Modem status register
#define EOI   0X20 // Palavra de controlo de fim de interrupcao
#define PEOI  0X20 // Porto para controlo de fim de interrupcao
#define PIEN  0X21 // Porto de autorizacao de interrupcoes

/***** Troca de mensagens *****/
// Delimitadores das mensagens
#define SON   0x60 // 28 Inicio de mensagem
#define SOM   0x64 // 28 Inicio de mensagem
#define SOT   0x6C // 28 Inicio de mensagem
#define SOF   0x68 // 28 Inicio de mensagem (identif. com 6bits)
#define EOT   0x3E // 29 Fim de mensagem

// Erros das rotinas transmissao e recepcao
#define NOERROR 0 // Transmissao valida
#define TXERROR 1 // Transmissao impossivel
#define RXERROR 2 // Recepcao impossivel
#define IDERROR 3 // Identificador
#define TOERROR 4 // Timeout
#define CMERROR 5 // Comando invalido
#define TOERR   0X03 // Timeout
#define CMERR   0X04 // Comando invalido
#define MSGSIZE 1023 // Maximo de caracteres a tx ou rx
typedef struct {
    unsigned int head; // Cabeça do buffer
    unsigned int tail; // Cauda do buffer
    unsigned char buf[MSGSIZE+1]; // Buffer de dados
}bufctinfo;
extern bufctinfo rx,tx; // Buffer de transmissao e recepcao

/* Variaveis que permitem identificar porta de comunicacao em utilizacao */
extern int comport; /*=COMPORT1;*/ // Endereco base */
extern int comportmen; /*=COMPORT1MEN;*/ // Autorizacao no 8259 */
extern int comportint; /*=COMPORT1INT;*/ // Interrupcao utilizada */
extern int comdly; // Timeout da comunicacao
extern int comerr; // Erros da comunicacao
extern unsigned int cont;

/***** Troca de comandos *****/
#define MAXCMD 255 // Numero maximo de caracteres de um comando
#define MAXTRY 3 // Numero maximo de retransmissoes
extern unsigned int /*existrx*//existrxcmd; // Sinaliza recepcao de uma mensagem
extern unsigned char txc[MAXCMD]; // Buffer do comando a transmitir
extern unsigned char first;

/***** Funcoes *****/
int initcom(char);
void restorestat(void);
int rxchr(unsigned char *chr);
int rxmsg(unsigned char msg[10],unsigned char *rxcmd);

```

```

/*****
/*      File : sensor5.c
/*      Description : Definição do modo de leitura do sensor e comunicações
/*      com o PIC e inicialização da configuração do PIC
/*      Compiler used : Borland C
/*      Written by: Antonio J. Barata
/*      Last revision: 07/2005
/*      Update & revision file: 07/2005
*****/
#include <pic.h>
#include <conio.h>
#include "delay.h"
#include <stdio.h>

/*****
/*  Declaracoes genericas
*****/
#define TRUE  1
#define FALSE 0
#define ON    1
#define OFF   0
#define TMR1PER      61470//475 // 888us - devia ser 61439
#define TMR1PERINI   63620//505 // 444us - devia ser 63487
#define TMR0PER      180 //30us
#define NPIXEL      1024 /* No de pixels do 'LIS-1024 */

/*****
/*  Declaracoes da comunicacao serie
*****/
#define TXIDLE  0 /* Transmissao nao activa */
#define TXBUSY  1 /* Transmissao a decorrer */
#define RXIDLE  2 /* Fora de recepcao */
#define RXBUSY  3 /* Em recepcao de mensagem */
#define RXDONE  4 /* Recepcao de mensagem realizada */
#define SON    0X60 /* Delimitador inicio de bloco - ultimo apos F*/
#define SOM    0X64 /* Delimitador inicio de bloco - ultimo de 1024*/
#define SOT    0X6C /* Delimitador inicio de bloco - 0x3E*/
#define SOF    0X68 /* Delimitador inicio de bloco */
#define EOT    0X3E /* Delimitador fim de bloco */
#define MAXBUFTX 80 /* Numero maximo de elementos do tx.buf */
#define MAXBUFRX 5  /* Numero maximo de elementos do rx.buf */

typedef struct { /* Controlo do estado de transmissao e recepcao */
    unsigned char stat; /* Estado da comunicacao */
    unsigned short n; /* Indice de tx/rx e numero de bytes */
    unsigned char buf[MAXBUFTX]; /* Buffer de dados */
} buftypetx;

typedef struct { /* Controlo do estado de transmissao e recepcao */
    unsigned char stat; /* Estado da comunicacao */
    unsigned short n; /* Indice de tx/rx e numero de bytes */
    unsigned char buf[MAXBUFRX]; /* Buffer de dados */
} buftyperx;

bank2 buftyperx rx;
bank2 buftypetx tx;
unsigned int rd2flg,rd3flg,adcflg,comutflg=0,comutflg1=0;
bit cicloenv,iniflg,flgfim,first,rxflg,flgend;
unsigned char sinc,ad1,ad2;
unsigned int naquis;//,ndata; /* no aquis=NPIXEL e no data=NFRAME*/

```

```

unsigned char adcbytel,adcbyteh,c,rxstat;
void Ini(void)
{
    /* option register */
    RBPU = 0; //PORTB pull-ups enabled
    INTEDG = 1; //interrupt on rising edge of RB0/INT pin
    /* Inibe todas as interrupcoes e limpa as flags */
    INTCON = 0X00;
    PIE1 = 0x00;
    PIR1 = 0x00;
    PIR2 = 0x00;

#if 0
    /* INTCON register*/
    GIE = 0; // disable all interrupts (A/D)
    PEIE = 0; // disable all peripheral interrupts (A/D)
    RBIF = 0; // None of the RB7:RB4 pins have changed state
#endif

    /* PIE1 register */
    ADIE = 0; // disable the A/D interrupt (A/D)
    /* PIE2 register */
    CCP2IE = 0; // disable the CCP2 interrupt
    /* PIR1 Register*/
    TMR2IF=0; // no TMR2 to PR2 match occurred (timer2)
    /* TRISx register ( configuração de entradas e saídas)*/
    TRISA = 0x0D; //001101
    TRISB = 0xCB; //11001011//11001010 0xCA antes
    TRISC = 0x80; //xx(10)000000
    RD2=0;
}

void PWM(void)
{
#if 0
    // INTCON registers
    GIE=0; //disable all interrupts (A/D)
    PEIE=0; // disable all peripheral interrupts (A/D)
#endif

    //PIR1
    TMR2IF=0; // no TMR2 to PR2 match occurred (timer2)
    //PEI2
    CCP2IE= 0; // disable the CCP2 interrupt
    /*T2CON timer2 control register*/
    TOUTPS2 = 0;
    TOUTPS1 = 0;
    TOUTPS0 = 0; //output postscaler select bits 1:1
    TMR2ON = 1; //Timer2 is on
    T2CKPS1 = 0;
    T2CKPS0 = 0; //clock prescaler select bits is 1 = 1x /// */
    PR2=0xFF; // Escolha do periodo PWM (max para 19200kb)
    CCP1L=0x80; //Escolha do Duty cycle ()
    CCP1X=1; // "
    CCP1Y=0; // "
    T2CON=0x06; // Activar TMR2 e Prescaler 16
    CCP1CON=0x0F; // configurar CCP1 como modulo PWM
}

```

```

void initimer0(void)
{
    /* option register */
    T0CS = 0; // internal instruction cycle clock (CLKOUT)
    PSA = 0; //prescaler is assigned to Timer0 module
    PS2=0; // prescaler 1:4 se frequencia 0 20MHz
    PS1=0;
    PS0=0;
    T0IF=0; //0=limpa a Flag
    T0IE=1; //existe interrupção do timer 0
    TMR0=TMR0PER; //valor inicial do timer
}

void adc (void)
{
    // A/D register
    ADCON1 = 0x1E; // refs do ADC VDD e VSS e ADFM a 1
    ADCON0 = 0x80; //10000000 configuração do modo de operação do A/D
}

void initimer1_delay(void)
{
    /* Escreve no contador */
    TMR1H = (TMR1PERINI & 0xFF00) >> 8; /* O byte mais significativo */
    TMR1L = (TMR1PERINI & 0x00FF); /* O byte menos significativo */
    TMR1IE = 1; /* Activa a interrupcao do timer */
    TMR1IF = 0; /* Limpa a flag */
    /* Prescaler de 1 e TIMER 1 OFF e incrementa com os ciclo m quina do PIC */
    T1CON = 0x00;
    /* Prescaler de 8 e TIMER 1 OFF e incrementa com os ciclo m quina do PIC */
    #if 0
        T1CON = 0x30;
    #endif
}

void ini_lis1024(void)
{
    /* Modo de leitura sincrona */
    RB5 = 0; //RST a 0 antes RB0 = 0
    RB2 = 1; //RMS a 1
    RB4 = 0; //PRE a 0
}

/*****
/* Permite as interrupções */
*****/
void enable_interrupt(void)
{
    INTCON = INTCON | 0xC0;
}

/*****
/* Inibe as interrupções */
*****/
void disable_interrupt(void)
{
    INTCON = INTCON & 0x3F;
}

/*****
/* Rotina de activacao das interrupcoes de transmissao e da recepcao */
*****/

```

```

/*****/
void acttxrxint(void)
{
    TXIE = 1;
}
/*****/
/* Rotina de autorizaç,ão da transmissao e da recepç,ao */
/*****/
void setutxerxe(void)
{
    TXEN = 1;
}
/*****/
/* Rotina de inicializac,ões */
/*****/
void iniu(void)
{
    TRISC7 = 1; /* Configura os pinos de rx e tx da UART do PIC */
    TRISC6 = 0;
    TXIE = 0; /* Inibe a interrupç,ão de transmissao */
    RCIE = 0; /* Inibe a interrupç,ão de recepç,ão */
    TXSTA = 0x04; /* Modo assincrono e 8 bits, inibe tx, BRGH=1 high speed*/
    RCSTA = 0x80; /* USART escolhida, inibe a recepç,ão */
    SPBRG = 19; /* Escolha do baud rate , 57600 com 18,432MHz BRGH=1*/
}
/*****/
/* Define tempo de espera */
/*****/
void delay(unsigned int del)
{
    unsigned int i,j;
    for(i=0;i<del;i++)
        for(j=0;j<del;j++);
}
/*****/
/* Rotina de transmissao de uma mensagem, com tempo de espera programado */
/*****/
int txmsg(void)//unsigned char txcmd //,unsigned char *txc,unsigned long nb)
{
    unsigned int cnt;//,i;
    cnt = 500; /* Inicializa contador de espera */
    while (tx.stat != TXIDLE && --cnt) /* Espera fim de transmissao */
        delay(1);
    if (cnt) {
        tx.stat = TXBUSY;tx.n = 0; /* Inicializa estado e indice */
        if (adcbytel==0x3E) {
            tx.buf[0] = SOT | adcbyteh; /* SOT qdo valor igual a EOT */
            tx.buf[1] = 0x3F;
        }
    }
    #if 0
    else if ((ad2==255) && (ad1!=3)){ /* incrementa ad1 qdo ad2=255*/
        ad1++;
        tx.buf[1] = ad2;
    }
    #endif
}

```

```

        else if (naquis==NPIXEL){ /* SOM qdo naquis=NPIXEL */
            tx.buf[0] = SOM | adcbyteh;
            tx.buf[1]= adcbytel;
        } else { /* SOF qdo valor normal */
            tx.buf[0] = SOF | adcbyteh; /* Inicio de mensagem */
            tx.buf[1]= adcbytel;
        }
        tx.buf[2] = EOT; /* Fim de mensagem */
    }
    return(cnt);
}
void main(void)
{
    disable_interrupt();
    ad1=0;
    ad2=255;
    Ini();
    ini_lis1024();
    iniu(); /* inicializa a USART */
    RC1=0;
    RC3=0;
    RC4=0;
    RC0=0;
    adc();
    initimer0();
    enable_interrupt();
    RCIE=1; /* enable interrupt de recepcao */
    CREN=1; /* enable recepcao de caracteres */
    adcfg = FALSE;
    iniflg = FALSE;
    rxflg=FALSE;
    first=TRUE;
    flgfim=FALSE;
    flgend=FALSE;
    cicloenv=FALSE;
    naquis = 0;
    while (1)
    {
        if (rxflg) { /* rotina apos receber qq tecla do PC - RXIF activo */
            RC4 = 0;
            rxflg = FALSE;
            rxstat = RCSTA;
            c = RCREG;
            if (iniflg == TRUE) {
                flgfim=TRUE;
            } else if (c == 'I' || c == 'i') { /* Detecta inicio de recepcao */
                iniflg=TRUE;
            }
            RCIE = 1; /* Enable das interrupcoes de recepcao */
        } /* fecha o rxflg */

        if (iniflg) { /* Inicio do programa apos receber tecla de inicio*/
            RC3 = 1;

            if (first) { /* 1a Vez que executa depois da tecla de inicio */
                first = FALSE;
                PWM();
            }
        }
    }
}

```



```

RC4=1;
#if 0
    if (naquis == NPIXEL) {
        naquis = 0;
        timer1off();
        while ((sinc = RB1) == 0);
            while ((sinc = RB1) == 1);
        initimer1_delay();
        timer1on();
    }
#endif
    initimer1_delay();
    TMR1ON = 1;
    ADON = 1;
} /* fecha o first */
if (adcflg) { /* rotina que se segue a conversao do ADC */
    adcflg = FALSE;
    RC3 = 0; //RC5
    RC1 = 1;
    if (flgfim == FALSE) { /* envia para o PC */
        txmsg();//,endram,endramant);
        TXIE = 1;
        TXEN = 1;
        RC1 = 0;
    } /* fecha o if flgfim==FALSE */
    if (naquis == NPIXEL) {
        naquis = 0;
        TMR1ON = 0;
        if (flgfim == FALSE) { /* se recebeu tecla de fim*/
#if 0
                if (cicloenvcnt==2) {
                    cicloenvcnt=0;
                } else cicloenvcnt++;
#endif
#endif
        //ENQUANTO NAO SE TEM O SENSOR - SYN NAO EXISTE
            while (sinc = RB1 == 0);
                while (sinc = RB1 == 1);

//TERMINA PARTE SEM SENSOR
                initimer1_delay();
                TMR1ON = 1;
            } /* fecha o if flgfim */
            else { /* executa se recebeu tecla */
                flgfim = FALSE;
                iniflg = FALSE;
                first = TRUE;
                ADON = 0; /* para o ADC */
            } /* fecha o else do if flgfim */
        } /* fecha o if naquis==NPIXEL */
    } /* fecha o if adcflg */
} /* fecha o iniflg */
} /* fecha o while(1) */
//T0IE=0;
} /* fecha o main */

/*****/

```

```

/* Atende as interrupções */
/*****/
static void interrupt intr(void)
{
    if(ADIF) {
        ADIF = 0;
        ADIE = 0;
        ADCON1 = 0x80; //adcch0off();
        RC3 = 1;//RB0
        naquis++;
        adcflg = TRUE;
    }

    if(TOIF && TOIE) {          /* Interrupcao do timer 0 */
        TOIF = 0;              /* Limpa a flag */
        TOIE = 0;
        RC4 = 0;//RB2
        ADGO = 1; //startconv();
        ADIE = 1; //enintadc();
    }

    if(TMR1IF) {              /* Interrupcao do timer 1 */
        TMR1IF = 0;           /* Limpa a flag */
        TMR1H = (TMR1PER & 0xff0) >> 8; /* O byte mais significativo */
        TMR1L = TMR1PER & 0x00FF;      /* O byte menos significativo */
        ADCON0 = 0x81; //adcch0on();
        RC4 = 1;//RB2
        TMR0 = TMR0PER; //define periodo de int do timer0
        TOIF = 0; //limpa flag do timer0
        TOIE = 1; //permite int do timer0 */
    }

    if(RCIF && RCIE) {
        rxflg = TRUE;
        RC4 = 1;
        RCIE = 0;
    }

    if(TXIF && TXIE) {
        TXREG = tx.buf[tx.n]; /* Envia caracter */
        if (tx.buf[tx.n] == EOT) { /* Testa fim mensagem */
            tx.stat = TXIDLE; /* Fim de transmissao */
            TXIE=0;
        }
        tx.n++;
    }

    #if 0
    if(TXIF && TXIE) {
        TXREG = tx.buf[tx.n]; /* Envia caracter */
        if (tx.buf[tx.n] == EOT) { /* Testa fim mensagem */
            tx.stat = TXIDLE; /* Fim de transmissao */
            TXIE=0;
        }
        tx.n++; txcnt++;
    }
#endif
}

```

```

/*****
/*      File : Conf-com.c                                          */
/*      Description : Inicialização do Micro, modo de leitura do sensor e          */
/*      configuração das comunicações do PIC com o PC pelo porto serie RS-232      */
/*      Description file: Sensor / comunicações /PIC                    */
/*      Compiler used : Borland C                                          */
/*      Written by: Antonio J. Barata                                     */
/*      Last revision: 09/2005                                           */
/*      Update & revision file: 09/2005                                   */
*****/

#include <dos.h>
#include <bios.h>
#include <time.h>
#include <math.h>
#include <conio.h>
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <process.h>
#include <io.h>
#include <fcntl.h>
#include <graphics.h>
#include "confocal.h"

int sysinit;
unsigned int cont;
unsigned int cont;

#if 0
unsigned int contador,contador1;
#endif

/*****
/*  Configuracao das comunicacoes                                          */
*****/
comecfginfo ccfg;

/*****
/*  Variaveis que permitem identificar porta de comunicacao em utilizacao  */
*****/
int comport; /*=COMPORT1;*/ /* Endereco base */
int comportmen; /*=COMPORT1MEN;*/ /* Autorizacao no 8259 */
int comportint; /*=COMPORT1INT;*/ /* Interrupcao utilizada */
unsigned char conf8259; /* Configuracao do controlador de interrupcoes */
unsigned char oldIER; /* Estado do registo de controlo de interrupcoes */
unsigned char oldMCR; /* Estado do registo de controlo de estado */
void interrupt (*vectCOM)(); /* Antigo vector de interrupcao de COM */
void interrupt handlerCOM(); /* Nova rotina de interrupcao de COM */

/*****
/*  Troca de mensagens                                                    */
*****/
bufctlinfo rx,tx; /* Buffer de transmissao e rececao */
#if TESTING
erretlinfo ec; /* Controlo de erros */
#endif
int comdly; /* Timeout da comunicacao */

```

```

int comerr;          /* Erros da comunicacao */

/*****
/* Troca de comandos
*****/
unsigned int existrxcmd; /* Sinaliza recepcao de uma mensagem */
unsigned char tmid ;

/***** Rotinas de comunicacao por interrupcao *****/
/*****
/* Rotina de inicializacao de comunicacoes
/* Devolve 1 se nao houve problema ou um valor negativo em caso contrario
*****/
int initcom(char porto)
{
    int stt;
    char comsett;
    switch(porto) {
        case '1':
            comport = COMPORT1;comportmen= COMPORT1MEN;comportint = COMPORT1INT;
            break;
        case '2':
            comport = COMPORT2;comportmen= COMPORT2MEN;comportint = COMPORT2INT;
        default:
            return(-1);
    }

    disable();          /* Nao autoriza interrupcoes */
    vectCOM = getvect(comportint); /* Guarda o vector de COM */
    setvect(comportint,handlerCOM); /* Instala o vector de COM */

#if 0
    ccfg.baudrate = BAUD9600;
    ccfg.databits = DATA8BIT;
    ccfg.stopbits = STOP1BIT;
    ccfg.parity = PARYNONE;
    comsett = ccfg.baudrate | ccfg.parity | ccfg.stopbits | ccfg.databits;
    bioscom(0, comsett, ccfg.serialport);
#endif

    outportb(comport+LCR,0X83); /*permite aceder DLR,no parity 1 stop 8 bits*/
    outportb(comport,BAUD57600); /* escreve LO byte */
    outportb(comport+IER,0); /* escreve HO byte */
    outportb(comport+LCR,0X03); /* nao permite aceder a DLR */

    oldIER = inportb(comport+IER); /* Antiga mascara para IRQ */
    outportb(comport+IER,ENRXRDY); /* Permite o receptor */
    conf8259 = inportb(PIEN); /* Guarda configuracao 8259 */
    outportb(PIEN,conf8259 & comportmen); /* Reprograma 8259 */
    oldMCR = inportb(comport+MCR); /* Mascara de linhas saida */
    outportb(comport+MCR,ENOUT2/* | SRTS | SDTR*/); /* Nova mascara, INT ENABLE */
    inportb(comport+DR); /* Limpa buffer da UART */
    inportb(comport+LSR); /* Limpa erros existentes */
    rx.head = rx.tail = 0; /* Inicializa ponteiros */
    tx.head = tx.tail = 0; /* Inicializa ponteiros */

#if TESTING
    ec.ntxc = 0; /* Limpa contadores */
    ec.ntxe = 0;
#endif

```

```

ec.nrx = 0;
ec.nsx = 0;
ec.fp = fopen("comm.ech","w");
#endif

comerr = NOERROR;          /* Limpa erro */
enable();                  /* Autoriza INT (hardware) */
return(1);
}

/*****
/* Rotina reposicao do estado inicial do computador */
*****/
void restorestat(void)
{
disable();                /* Nao autoriza INT */
outportb(PIEN,conf8259);  /* Repoe do 8259 */
setvect(comportint,vectCOM); /* Instala antigo vector de COM */
outportb(comport+IER,oldIER); /* Antiga mascara de INT4 */
outportb(comport+MCR,oldMCR); /* Antiga mascara de linhas */
inport(comport+DR);      /* Limpa buffer da USART */

#if 0
inportb(comport+DR);      /* Limpa buffer da UART */
inportb(comport+DR);      /* Limpa buffer da UART */
#endif

inport(comport+LSR);      /* Limpa erros existentes */
enable();                 /* Autoriza interrupcoes */

#if TESTING
if(ec.fp != NULL) {
fprintf(ec.fp,"\r\nCmnds: %d, Txe: %d, Rxe: %d, Sxe: %d",ec.ntxc,ec.ntxe,ec.nrx,ec.nsx);
fclose(ec.fp);
}
#endif
}

/*****
/* Interrupt handler das comunicacoes (COM) */
*****/
void interrupt handlerCOM()
{
unsigned char c;
disable();
if (((stat=inportb(comport+IIR)) & 0X02) && !(stat & 0x06)) { /* Interrupcao
transmissor */
c = tx.buf[tx.tail];
cont++;
outportb(comport+DR,c);
tx.tail = (tx.tail+1) & MSGSIZE; /* Buffer circular */
if(tx.tail == tx.head) { /* Testa fim mensagem */
outportb(comport+IER,ENRXRDY); /* Activa rx */
}
}
}
}

```

```

else {
    cont++;
    rx.buf[rx.head] = c = inportb(comport+DR);
    rx.head = (rx.head + 1) & MSGSIZE;    /* Buffer circular */
}

#if 0
    if (c == 0x67/*EOT*/) {
        existrxcmd=TRUE;    /* Sinaliza rx mensagem */
    }
#endif
}
outportb(PEOI,EOI);    /* Fim de interrupcao 8259 */
enable();
}

/*****
/* Rotina de rececao de um caracter
*****/
int rxchr(unsigned char *chr)
{
    unsigned int cnt,j/*,inc=0*/;
    cnt = 200;
    while ((rx.head == rx.tail) && (--cnt));    // Espera caracter
    if (cnt) {
        disable();
        *chr = rx.buf[rx.tail];    /* Caracter recebido */
        rx.tail = (rx.tail+1) & MSGSIZE;    /* Buffer circular */
        enable();
        comerr = NOERROR;    /* Rececao correcta */
    }
    else {
        comerr = TOERR;    /* Erro de timeout */
    }
    return(comerr);
}

/*****
/* Rotina de rececao de uma mensagem, com tempo de espera programado */
/* Novo identificador SOF de 6 bits
*****/
int rxmsg(unsigned char msg[10],unsigned char *rxcmd/***rxc*/)
{
    struct text_info stxt;
    unsigned char rmid;
    unsigned char c, nump;
    int n,j,cnt;
    int schkrx,b;
    int ind;
    int i=0;
    cnt = 6;    // d tempo para execucao do comando pela UR (cnt*200ms)
    do {
        if(rxchr(&c) != NOERROR) {    // espera 200ms
            if (--cnt <= 0) {
                comerr = TOERROR;
            }
        }
    } while((c & 0xF0) != 0x60);
    msg[i++] = c; //TENTATIVA DE REDUZIR TEMPO // Copia SOT
}

```

```

do {
    if(rxchr(&c) != NOERROR) {
        comerr = TOERROR;
        return(comerr);
    }
    msg[i++] = c;          // Copia bytes recebidos
} while(c != EOT);

*rxcmd = msg[0] & 0xFC; //TENTATIVA DE REDUZIR TEMPO

#if 0
if((msg[0] & 0x04) {
    msg[0]=0x80;
    msg[1]=0x3E;
} else msg[0] = msg[0] & 0x03;
#endif
comerr = NOERROR;
return(comerr);
}

#if 0
/*****
/* Rotina de recepcao de uma mensagem, com tempo de espera programado */
*****/
int rxmsg(unsigned char *rxcmd,unsigned char *rx, unsigned int *nb)
{
    struct text_info stxt;
    unsigned char rmid;
    unsigned char msg[MSGSIZE];
    unsigned char c, nump;
    int n,i,/*j=0,*/cnt;
    int schkrx,b;
    int ind;
    do {
        if(rxchr(&c) != NOERROR) { // espera 200ms
            if(--cnt <= 0) {
                comerr = TOERROR;
                return(comerr); /* deixa de haver erros de timeout */
            }
        } while(c != SOT);
        msg[0] = c;          // Copia SOT
        i = 1;
        do {
            if(rxchr(&c) != NOERROR) {
                comerr = TOERROR;
                return(comerr);
            }
            msg[i++] = c;          // Copia bytes recebidos
        } while(c != EOT);

        existrx--;          // Sinaliza fim da mensagem
        *rxcmd = msg[1];    // Comando recebido
        *nb = 2;           // Numero de bytes comando

        for(ind=0;ind<*nb;ind++)
            *(rx+c+ind)=msg[2+ind];

```

```

#if 0
    n = dcodmsg(msg+4,rx,*nb) + 6;          // Converte a binario
    /* n = número de parametros codificados recebidos + 6 = número */
    /* total de bytes recebidos */
    schkrx = 0X00;                          /* inicializa sumcheck */
    for(b = 1;b < n-2;b++)                    /* calcula sumcheck */
        schkrx ^= msg[b];
    schkrx = schkrx | 0X80;
    if( schkrx != msg[n-2])
        return(TOERROR); /* se o xor não for correcto avisa que existe erro */

    if(i == n-1 || i == n) {
        comerr = NOERROR;
    }

#if TESTING
    ftime(&etb);
    if(ec.fp != NULL)
        fprintf(ec.fp, "\r\n{Cmd:%02x - Dur:%d}",msg[5],
            etb.time*1000+etb.millitm-(btb.time*1000+btb.millitm));
#endif
    }
    else
        comerr = CMERROR;
#endif
    comerr = NOERROR; /* acrescentei agora */

/* apos retirar o if rmid=tmid */
#if 0
    }
    else
        comerr = IDERROR;
#endif

#if 0
    }
#endif
    return(comerr);
}
#endif

/***** Rotinas auxiliares teclado / menus / edicao *****/
/*****
/* Termina programa */
/*****
void termina(void)
{
#if 0
    if(ccfg.config == REMCONN && connct)
        disconnect();
#endif
    window(1,1,80,25);textcolor(WHITE);textbackground(BLACK);
    _setcursortype(_NORMALCURSOR);
    lowvideo();
    clrscr();
    restorestat();
    exit(0);
}

```



```

/*****
/* Processa os erros irrecuperaveis                               */
/*****
void fatalerror(char *m1,char *m2)
{
    char ind,buffer[500];
    char ch[]={188,(RED << 4) | WHITE};
    int c;
    soundwarning(100);
    gettext(34,14,74,19,buffer);
    window(34,14,74,19);textbackground(RED);textcolor(WHITE);
    clrscr();
    putch(201);
    for (ind = 0;ind < 39;++ind)
        putch(205);
    putch(187);
    for (ind = 2;ind < 6;++ind) {
        gotoxy(1,ind); putch(186);
        gotoxy(41,ind);putch(186);
    }
    gotoxy(1,6);putch(200);
    for (ind = 0;ind < 39;++ind)
        putch(205);
    puttext(74,19,74,19,ch);
    gotoxy(4,2);cprintf(m1);
    gotoxy(4,3);cprintf(m2);
    gotoxy(11,5);cprintf("Enter - Termina");
    do {
        c = toupper(getch());
    }while(c != RET);
    puttext(34,14,74,19,buffer);
    window(1,1,80,25);textbackground(BLUE);textcolor(YELLOW);
    termina();
}

/*****
/* Rotina que emite um som de aviso                               */
/*****
void soundwarning(int sd)
{
    delay(sd);
}

/***** Gestao do ficheiro de configuracao *****/
/*****
/* Leitura de um parmetro do ficheiro de configuraçao          */
/*****
int rdcfgprm(char *cfgfile,long *fpos,char *prmname,char *prm)
{
    int n;
    char prmnamecod[MAXPRMLEN],buf[MAXPRMLEN];
    FILE *fp;

    if((fp = fopen(cfgfile,"r")) == NULL) // Abre ficheiro configuracao
        return(0);
    if(*fpos != 0)
        fseek(fp,*fpos,SEEK_SET);
}

```

```

strcpy(prmnamecod,"$");
strcat(prmnamecod,prmname);
strcat(prmnamecod,"$");
do
  n = fscanf(fp," %100s",buf);
while(strcmp(buf,prmnamecod) && n != -1);
if(n == -1) {
  *fpos = 0;
  fclose(fp);
  return(-1);
}
*fpos = ftell(fp)-100L;
fscanf(fp," %100s",prm);
fclose(fp);
if(prm[0] == '$')
  strcpy(prm,"");
return(TRUE);
}

/*****
/* Leitura da informacao de configuracao das comunicacoes */
/* Devolve 1 se nao houve problema ou um valor negativo em caso contrario */
*****/
int rdcomcfg(char *cfgfile, comcfginfo *ccpt)
{
  int stt;
  long fpos;
  char buf[MAXPRMLEN];
  fpos = 0L;
#ifdef 0
  if ((stt = rdcfgprm(cfgfile, &fpos, "ConnectionType", buf)) != TRUE)
    return(stt);
  else if (!strcmp(buf, "MODEM"))
    ccfg.config = REMCONN;
  else {
    if (!strcmp(buf, "RADIO"))
      ccfg.config = MDMCONN;
    else
      ccfg.config = LOCCONN;
  }
  if ((stt = rdcfgprm(cfgfile, &fpos, "PwdLev", buf)) != TRUE)
    return(stt);
  else
    gcfg.pwdlev = strcmp(buf, "ÿISAÿ") ? strcmp(buf, "Quantific") ? 0 : 1 : 2;
#endif
  if((stt = rdcfgprm(cfgfile,&fpos,"SerialPort",buf)) != TRUE)
    return(stt);
  else if(!strcmp(buf, "COM1"))
    ccpt->serialport = COM1;
  else if(!strcmp(buf, "COM2"))
    ccpt->serialport = COM2;
  else if(!strcmp(buf, "COM3"))
    ccpt->serialport = COM3;
  else if(!strcmp(buf, "COM4"))
    ccpt->serialport = COM4;
  else
    return(-1);
}

```

```
if((stt = rdcfgprm(cfgfile,&fpos,"IRQNumber",buf)) != TRUE)
    return(stt);
else if(!strcmp(buf,"IRQ3"))
    ccpt->irqnumber = IRQ3;
else if(!strcmp(buf,"IRQ4"))
    ccpt->irqnumber = IRQ4;
else
    return(-1);

if((stt = rdcfgprm(cfgfile,&fpos,"BaudRate",buf)) != TRUE)
    return(stt);
else if(!strcmp(buf, "110")) {
    ccpt->baudrate = BAUD110;
    comdly = 64*150/110;
}
else if(!strcmp(buf, "150")) {
    ccpt->baudrate = BAUD150;
    comdly = 64;
}
else if(!strcmp(buf, "300")) {
    ccpt->baudrate = BAUD300;
    comdly = 32;
}
else if(!strcmp(buf, "600")) {
    ccpt->baudrate = BAUD600;
    comdly = 16;
}
else if(!strcmp(buf, "1200")) {
    ccpt->baudrate = BAUD1200;
    comdly = 8;
}
else if(!strcmp(buf, "2400")) {
    ccpt->baudrate = BAUD2400;
    comdly = 4;
}
else if(!strcmp(buf, "4800")) {
    ccpt->baudrate = BAUD4800;
    comdly = 2;
}
else if(!strcmp(buf, "9600")) {
    ccpt->baudrate = BAUD9600;
    comdly = 1;
}
else
    return(-1);

if((stt = rdcfgprm(cfgfile,&fpos,"DataBits",buf)) != TRUE)
    return(stt);
else if(!strcmp(buf, "7"))
    ccpt->databits = DATA7BIT;
else if(!strcmp(buf, "8"))
    ccpt->databits = DATA8BIT;
else
    return(-1);          // erro no par/metro

if((stt = rdcfgprm(cfgfile,&fpos,"StopBits",buf)) != TRUE)
```

```
    return(stt);
if(!strcmp(buf, "1"))
    ccpt->stopbits = STOP1BIT;
else if(!strcmp(buf, "2"))
    ccpt->stopbits = STOP2BIT;
else
    return(-1);          // erro no par/metro

if((stt = rdcfgprm(cfgfile,&fpos,"Parity", buf)) != TRUE)
    return(stt);
else if(!strcmp(buf, "NONE"))
    ccpt->parity = PARYNONE;
else if(!strcmp(buf, "ODD"))
    ccpt->parity = PARYODD;
else if(!strcmp(buf, "EVEN"))
    ccpt->parity = PARYEVEN;
else
    return(-1);          // erro no par/metro
#endif
/* Atraso nas comunicações */
delaycom = 300;        /* Atraso entra a emissão e a recepção */
delaycmd = 30;         /* Atraso entre comandos */
if((stt = rdcfgprm(cfgfile,&fpos,"Delrxtx", buf)) != TRUE)
    return(-1);
delaycom = atoi(buf);
if((stt = rdcfgprm(cfgfile,&fpos,"Delcmds", buf)) != TRUE)
    return(-1);
delaycmd = atoi(buf);
#endif
return(1);
}
```

```

/*****
/*      File : Confocal.c                               */
/*      Description : Programa para a visualização gráfica dos resultados  */
/*      obtidos do sensor no computador                               */
/*      Compiler used : Borland C                               */
/*      Written by: Antonio J. Barata                               */
/*      Last revision: 09/2005                               */
/*      Update & revision file: 09/2005                               */
*****/

/*****
/* Library functions utilizadas */
*****/
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <math.h>
#include <dos.h>
#include <io.h>
#include <process.h>
#include <fcntl.h>
#include <ctype.h>
#include <time.h>
#include "confocal.h"

/*****
/*      Resumo das funcoes usadas                               */
*****/
void      ftempo(void),
          reset(void),
          opengraph(void),
          apresentacao(void),
          apresentacao1(void),
          grafico(void),
          apaga(void),
          stop(void),
          shell(void),
          menu_de_ajudas(int a),
          mover_menu(void),
          selecao(int a),
          funcao_escolhida(int a),
          funcao_sample_rate(void),
          escala_cima(void),
          graf_ampli_cima(void),
          escala_baixo(void),
          graf_ampli_baixo(void),
          coord_xx_cima(float ampli_max,float ampli_min),
          coord_xx_baixo(float ampli_max,float ampli_min);

unsigned int ler_adc(void);
float *vector(int nl,int nh);
void free_vector(float *v,int nb,int nh);

/*****
/* Definicao das constantes utilizadas */
*****/
#define RETURN 13

```

```

#define ESC 0x1B
#define BK 0x0d
#define stat1 0x305 /*endereço de registo de texto*/
#define stat2 0x306 /*endereço de registo de texto*/
#define FF1 0x301 /*endereço do FIFO1*/
#define FF2 0x302 /*endereço do FIFO2*/
#define FF3 0x303 /*endereço do FIFO3*/
#define FF4 0x304 /*endereço do FIFO4*/
#define NDAT 1024
#define NVAL 8
#define MDAT 448
#define UP 0x48 // seta para cima
#define DOWN 0x50 // seta para baixo
#define LEFT 0x4B // seta para a esquerda
#define RIGHT 0x4D // seta para a direita

/*****
/* Declaracao de variaveis */
*****/
int tecla, escolha, sair, sample_rate, nlinhas, ncolunas, i;
unsigned int adc,bufadc[NDAT],/* datx[NDAT], daty[NDAT],*/ conf, mediaadc,nbytes;
unsigned long total,total2;

float /*results[NDAT],*/media,media2,/*total,xmax,ymax,*/
ampli_min_cima,ampli_min_baixo,dx_cima,dx_baixo,
min_ant_cima, min_ant_baixo, max_ant_cima, max_ant_baixo,
ampli_max_cima, ampli_max_baixo, ampli_max, ampli_min;
int xmax,ymax;
char fich[20], fich1[20], fich2[20], valor1[25];
unsigned char first;
struct time timestr;
unsigned char timeinisec,timeinimin,timeanthour;
unsigned char timesec,timemin,timehour;
float *mediav;
int namost;
char porto;

/*****
/* Funcao que inicia aquisicao de dados do portati */
*****/
void ftempo_port() {
int i, j, x, y,bufcont[MSGSIZE], p;
unsigned long int atraso;
unsigned int m1, valor1,valor2, bsig;
char valor[25];
char valo[25];
float valpixel;
unsigned char c,aquis;
FILE *fp1,*fp2,*fp;
unsigned char *rxcmdd,rxch[3];
unsigned int *nb;
unsigned char adcvall,adcvalh,cmd;
unsigned int ind=0,indf,val=0,nscan=0;
unsigned int cyclecnt=0;

first=TRUE;
total2=0;
existrxcmd=FALSE;

```

```

aquis=FALSE;
ampli_min_cima = 0;
ampli_max_cima = MDAT;
ampli_min_baixo = 0;
ampli_max_baixo = NDAT;
dx_cima = 500 / (ampli_max_cima - ampli_min_cima);
dx_baixo = 500 / (ampli_max_baixo - ampli_min_baixo);
coord_xx_baixo(ampli_max_baixo,ampli_min_baixo);
setfillstyle(SOLID_FILL, BLACK); /* LIMPA GRAFICOS */
bar(13, 2/*203/*2 * ymax / 5 + 11*/ , 524/*588/*(9 * 64) + 12*/ , 369/* 4 * ymax / 5 - 4*/);
setfillstyle(SOLID_FILL, EGA_BLUE); /*LIMPA ESCRITA DA JANELA AZUL*/
bar(13, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);

#if 0
    j=0;
    total=0;
    contador=contador1=cont=0;
#endif

if ((fp = fopen(fich, "w+")) == NULL) {
    printf("Erro!");
    exit(0);
}

if(initcom(porto) != 1)
fatalerror("Erro de inicializaç,,"o","Ficheiro de configuraç,,"o inv lido");
menu_de_ajudas(13);
menu_de_ajudas(17);

do {
    if (kbhit())
        c = getch();
} while (c != 'T' && c != 'I');
tx.buf[tx.head]= c;
tx.head = (tx.head+1) & MSGSIZE;
outportb(comport+IER,(ENRXRDY|ENTXRDY));
menu_de_ajudas(10);
menu_de_ajudas(16);
*rxcmdd=0; /* inicializa para nao ser igual a SON */

do
{
    if((rx.head >= rx.tail + 3) || ((rx.tail > rx.head) && (rx.tail <= rx.head + MSGSIZE - 3))){
        cyclecnt=0;
        if(rxmsg(rxch,rxcmdd) == NOERROR) {
            aquis=TRUE; /* le msg serie */
        }
        else {
            menu_de_ajudas(15);
            val++;
        }
    }
    if(aquis) { /* nao funciona com esta flag */
        aquis = FALSE;
    }
}

#if 0
    do {
        rxmsg(rxch,rxcmdd);
        adcvilh = rxch[0] & 0x03;
        if (*rxcmdd == SOT) {

```

```

        adcvall = 0x3E;
    } else if (*rxcmdd == SON) {
        adcvall = 0x67;
    } else if ((*rxcmdd == SOF)||(*rxcmdd == SOM)) {
        adcvall = rxch[1];
    }
    bufadc[ind++] = (adcvall << 8) + adcvall;
} while (*rxcmdd != SOM);
#endif

adcvallh = rxch[0] & 0x03;
if (*rxcmdd == SOT) {
    adcvall = 0x3E;
} /*else if (*rxcmdd == SON) {
    adcvall = 0x67;
} */else if (*rxcmdd == SOF){
    adcvall = rxch[1];
} else if ((*rxcmdd == SOM)||(*rxcmdd == SON)) {
    adcvall = rxch[1];
}

#if 0
else if (*rxcmdd == SOF) {
    adcvall = rxch[1];
} else if (*rxcmdd == SOM) {
    adcvall = rxch[1];
    existrxcmd = FALSE;
}
#endif

bufadc[ind++] = (adcvallh << 8) + adcvall;
if ((*rxcmdd == SOM)||(*rxcmdd == SON)) {
    setfillstyle(SOLID_FILL, BLACK);
    bar(13, 2/*203 2 * ymax / 5 + 11*/ , 524 /*588 (9 * 64) + 12*/ , 369/* 4 * ymax
/ 5 - 4*/);

    p=0;
    for (i=0;i<ind;i++) {
#if 0
        x= bufadc[i] >> 3; /* espacio vertical pequeno */
        putpixel(p+13, 178/*92=2 * ymax / 5 - 100*/ - x , CYAN);
    } else
    {
#endif
        x= bufadc[i] >> 1; /* espacio vertical pequeno */
        putpixel(p+13/*-435*/*MDAT+13*/ , 350/*383=4 * ymax /
5*/ - x + 150, CYAN);

        p++;i++;
    } //fecha o ciclo for
    indf=ind;
    ind=0;
    cyclecnt = 0;

    if (/*(cyclecnt == 1) &&*/ first) {
        setfillstyle (SOLID_FILL, EGA_BLUE);
        bar(13, 403/*4*ymax/5+20*/ , 320/*xmax/2*/ , 448/*ymax-
2*ymax/40-7*/);

        menu_de_ajudas(14);
        first = FALSE;
    }
}

```



```

        setfillstyle (SOLID_FILL, EGA_BLUE);
        bar(175, 420, 320, 430);
        sprintf(valo, " %d", ++nscan);
        sprintf(valor, " %d", bufadc[512]);
        setcolor(WHITE);
        outtextxy(180, 424, "n§ scan: ");
        outtextxy(220, 424, valo);
        outtextxy(300, 424, valor);
    } //fecha o if *rxcmdd

#if 0
    if (/*(cyclecnt == 1) &&*/ first) {
        setfillstyle (SOLID_FILL, EGA_BLUE);
        bar(13, 403/*4*yymax/5+20*/, 320/*xmax/2*/, 448/*yymax-2*yymax/40-7*/);
        menu_de_ajudas(14);
        first = FALSE;
    }
#endif
#if 0
    if (cyclecnt == 1024) {
        setfillstyle (SOLID_FILL, EGA_BLUE);
        bar(180, 420, 280, 430);
        sprintf(valor, " %5.0f", media);
        setcolor(WHITE);
        outtextxy(190, 425, "MDI A : ");
        outtextxy(255, 425, valor);
        cyclecnt = 0;
    } /* fecha cyclecnt - 1024 dados recebidos */
#endif
    } /* fecha o if aquis - msg recebida sem erros*/

    } /* fecha o if rx.head >= rx.tail + 3 ...*/
    else if (cyclecnt++ > 20000) /* fecha o if existrx - msg recebida */
        menu_de_ajudas(16);
        cyclecnt=0;
        first=TRUE;
        rx.head = rx.tail = 0; /* Inicializa ponteiros */
    }

#if 0
    if (kbhit()) {
        c = getch();
        tx.buf[tx.head]= c;
        tx.head = (tx.head+1) & MSGSIZE;
        outputb(comport+IER,(ENRXRDY|ENTXRDY));
    }
#endif

} while(!kbhit())/*&&(*rxcmdd!=SON)*/; /* so para com tecla */
    c = getch();
    tx.buf[tx.head]= c;
    tx.head = (tx.head+1) & MSGSIZE;
    outputb(comport+IER,(ENRXRDY|ENTXRDY));

for (i=0;i<indf;i++)
    fprintf(fp, " %d\n", bufadc[i]*rxch[i]*);
restostat();

```

```

        menu_de_ajudas(9);
        fclose(fp);
    }
    /***** fim ftempo_port() *****/

    /*****
    /*      modo grafico      */
    /*****
void opengraph(){
int gdriver = DETECT, gmode, errorcode;
/*  struct text_info info; */
    directvideo = 0;
    detectgraph(&gdriver, &gmode);
    if (gdriver < 0) {
        printf("Hardware grafico nao detectado !\n");
        exit(1);
    }
    initgraph(&gdriver, &gmode, "");
    errorcode = graphresult();
    if (errorcode != grOk) {
        printf("initgraph error : %s.\n", grapherrormsg(errorcode));
        getch();
        exit(1);
    }
}

    /***** fim opengraph() *****/

    *****/
    * funcao que desenha o menu_principal *
    *****/
void apresentacao() {
    setfillstyle(SOLID_FILL, EGA_CYAN);
    bar(555, 1, 638, 5 * ymax / 20 + 19); //menu a direita
    bar(13, ymax - 2 * ymax / 40 + 1, xmax - 1, ymax - 1); //baixo
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(13, 4 * ymax / 5 + 40, xmax - 1, ymax - 2 * ymax / 40 - 7);
    setcolor(EGA_WHITE);
    rectangle( 554, 0, 639, 5 * ymax / 20 + 20); //menu a direita
    rectangle(12, 4 * ymax / 5 + 19, xmax, ymax - 2 * ymax / 40 - 6); //Iniciar... - em baixo
    rectangle(554, 145, 639, 190); //porto serie
    /*rectangle(12, 0, xmax - xmax / 5 + 1, 3 * ymax / 4);*/
    rectangle(12, ymax - 2 * ymax / 40, xmax, ymax); //baixo
    setcolor(EGA_BLACK);
    seleccao(1);
    seleccao(2);
    seleccao(3);
    seleccao(4);
    seleccao(5);

    #if 0 //CONFOCAL
        setcolor(EGA_WHITE);
        outtextxy(xmax - xmax / 10 + 10, 6 * ymax / 20 + 8 + 20 + 20, "Sample Rate");
        outtextxy(xmax - xmax / 10 + 10, 6 * ymax / 20 + 8 + 20 + 35, "grafico cima");
        outtextxy(xmax - xmax / 10 + 10, 6 * ymax / 20 + 8 + 20 + 50, " 2s  ");
        setcolor(RED);
        outtextxy(xmax - xmax / 10 + 10, 6 * ymax / 20 + 8 + 20 + 80, "Sample Rate");
        outtextxy(xmax - xmax / 10 + 10, 6 * ymax / 20 + 8 + 20 + 95, "grafico baixo");
    #endif
}

```

```

        outtextxy(xmax - xmax / 10 + 10, 6 * ymax / 20 + 8 + 20 + 110, " 200ms ");
#endif
    setfillstyle(SOLID_FILL, EGA_CYAN);
    bar(555, 146, 638, 189);
    setcolor(BLACK);
    outtextxy(597, 160, "Porto");
    if (porto=='1')
        outtextxy(597, 175, "s,rie:COM1");
    else
        outtextxy(597, 175, "s,rie:COM2");
    grafico();
}
/***** fim apresentacao() *****/

/*****
* funcao que desenha a janela inicial *
*****/
void apresentacao1() {
    char f[20];
    setfillstyle(SOLID_FILL, EGA_CYAN);
    bar(xmax / 5, 0, 4*xmax/5, ymax/5-10);
    outtextxy(xmax/5 + 20, ymax / 10 - 10, "MICROSCOPIA CONFOCAL");
    outtextxy(xmax/5 + 20, ymax / 10 + 10, "SENSOR - LIS-1024");
    setfillstyle(SOLID_FILL, EGA_CYAN);
    bar(xmax / 5, 4*ymax/5+10, 4*xmax/5, ymax);
    outtextxy(xmax/5 + 20, 9*ymax / 10 - 10, "CONFIGURACAO INICIAL DO PORTO SERIE");
    outtextxy(xmax/5 + 20, 9*ymax / 10 + 10, "COM2 (57600,8,1,N)");
    setfillstyle(SOLID_FILL, EGA_BLUE);
    setcolor(EGA_WHITE);
    rectangle(xmax / 5-10, ymax/5, 4*xmax/5+10, 4*ymax/5);
    bar(xmax / 5-10, ymax/5, 4*xmax/5+10, 4*ymax/5);
    setcolor(EGA_WHITE);
    outtextxy(xmax/5 + 20, 2*ymax / 5 + 10, "Nome dos ficheiros a criar ");
    outtextxy(xmax/5 + 20, 2*ymax / 5 + 37, "(maximo de 6 letras:)");
    setfillstyle(SOLID_FILL, EGA_BLACK);
    bar(xmax / 5 + 200, 2*ymax/5+30, xmax/5 + 250, 2*ymax/5 + 50);
    gotoxy(42, 15);
    f[0]=7;
    cgets(f);
    for(i=0; i<f[1]; i++) fich[i] = f[i+2];
    fich[i]='.';
    fich[i+1]='t';
    fich[i+2]='x';
    fich[i+3]='t';
    fich[i-4]=f[i-2];
}

/*****
/* Funcao que desenha o plano do grafico. Faz a divisao do espaco para */
/* apresentar o dos valores em baixo e medias em cima */
*****/
void grafico() {
    setcolor(EGA_WHITE);

#if 0
    rectangle(12, 0, (7 * 64) + 13, 2 * ymax / 5 - 10);
#endif

```

```

line(1 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 13, 1 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 7);
line(2 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 13, 2 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 7);
line(3 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 13, 3 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 7);
line(4 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 13, 4 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 7);
line(5 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 13, 5 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 7);
line(6 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 13, 6 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 7);
line(12, 2 * ymax / 5 - 13, 12, 2 * ymax / 5 - 7);
line(7 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 13, 7 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 7);

#endif
rectangle(12, 1/*2 * ymax / 5 + 10*/ , 525/*(9 * 64) + 13*/ , 370/*4 * ymax / 5*/);
line(1 * 513 / 9 + 12, 370/*4 * ymax / 5 - 3*/ , 1 * 513 / 9 + 12, 373/*4 * ymax / 5 + 3*/);
line(2 * 513 / 9 + 12, 370/*4 * ymax / 5 - 3*/ , 2 * 513 / 9 + 12, 373/*4 * ymax / 5 + 3*/);
line(3 * 513 / 9 + 12, 370/*4 * ymax / 5 - 3*/ , 3 * 513 / 9 + 12, 373/*4 * ymax / 5 + 3*/);
line(4 * 513 / 9 + 12, 370/*4 * ymax / 5 - 3*/ , 4 * 513 / 9 + 12, 373/*4 * ymax / 5 + 3*/);
line(5 * 513 / 9 + 12, 370/*4 * ymax / 5 - 3*/ , 5 * 513 / 9 + 12, 373/*4 * ymax / 5 + 3*/);
line(6 * 513 / 9 + 12, 370/*4 * ymax / 5 - 3*/ , 6 * 513 / 9 + 12, 373/*4 * ymax / 5 + 3*/);
line(7 * 513 / 9 + 12, 370/*4 * ymax / 5 - 3*/ , 7 * 513 / 9 + 12, 373/*4 * ymax / 5 + 3*/);
line(8 * 513 / 9 + 12, 370/*4 * ymax / 5 - 3*/ , 8 * 513 / 9 + 12, 373/*4 * ymax / 5 + 3*/);
line(12, 370/*4 * ymax / 5 - 3*/ , 12, 373/*4 * ymax / 5 + 3*/);
line(525, 370/*4 * ymax / 5 - 3*/ , 525, 373/*4 * ymax / 5 + 3*/);
line(525, 1 * 350 / 7/*4 * ymax / 5 - 3*/ , 527, 1 * 350 / 7/*4 * ymax / 5 + 3*/);
line(525, 2 * 350 / 7/*4 * ymax / 5 - 3*/ , 527, 2 * 350 / 7/*4 * ymax / 5 + 3*/);
line(525, 3 * 350 / 7/*4 * ymax / 5 - 3*/ , 527, 3 * 350 / 7/*4 * ymax / 5 + 3*/);
line(525, 4 * 350 / 7/*4 * ymax / 5 - 3*/ , 527, 4 * 350 / 7/*4 * ymax / 5 + 3*/);
line(525, 5 * 350 / 7/*4 * ymax / 5 - 3*/ , 527, 5 * 350 / 7/*4 * ymax / 5 + 3*/);
line(525, 6 * 350 / 7/*4 * ymax / 5 - 3*/ , 527, 6 * 350 / 7/*4 * ymax / 5 + 3*/);
line(525, 350/*4 * ymax / 5 - 3*/ , 527, 350/*4 * ymax / 5 + 3*/);
}
/***** fim grafico() *****/

void apaga()
{
    setfillstyle(SOLID_FILL, EGA_BLACK);
    bar(0,0, xmax, ymax);
    apresentacao();
    menu_de_ajudas(3);
}
/***** fim apaga() *****/

void stop(void)
{
    menu_de_ajudas(3);
    menu_de_ajudas(8);
}
/***** fim stop() *****/

void configurar(void)
{
    menu_de_ajudas(11);
    do {
        porto=getch();
    } while ((porto!='1')&&(porto!='2'));
    menu_de_ajudas(3);
    menu_de_ajudas(12);
}
/***** fim configurar() *****/

```

```

void shell(void)
{
    closegraph();
    printf("\n\nType EXIT to return to acquisition software\n");
    system("c:\\command.com");
    opengraph();
    clearviewport();
    setbkcolor(EGA_BLACK);
    settxtjustify(CENTER_TEXT, CENTER_TEXT);
    apresentacao();
    apaga();
    menu_de_ajudas(3);
    menu_de_ajudas(8);
}
/***** fim shell() *****/

/*****
/* Funcao que contem o menu das informacoes dadas ao utilizador */
/* ao longo do programa */
*****/
void menu_de_ajudas(int a) {
    char seta1[80], seta2[80];
    switch (a) {
    case 1:
        sprintf(seta1, "%c", 24);
        sprintf(seta2, "%c", 25);
        setfillstyle(SOLID_FILL, EGA_CYAN);
        bar(13, ymax - 2 * ymax / 40 + 1, xmax - xmax / 5, ymax - 1);
        setcolor(EGA_RED);
        outtextxy(1 * xmax / 5 - 20, ymax - (ymax / 26) / 2, seta1);
        outtextxy(1 * xmax / 5 - 10, ymax - (ymax / 26) / 2, seta2);
        outtextxy(3 * xmax / 5 - 50, ymax - (ymax / 26) / 2, "<<ENTER>>");
        setcolor(EGA_WHITE);
        outtextxy(1 * xmax / 5 + 30, ymax - (ymax / 26) / 2, "MOVER");
        outtextxy(3 * xmax / 5 + 45, ymax - (ymax / 26) / 2, "ESCOLHER OPCAO");
        break;
    case 2:
        sprintf(seta1, "%c", 27);
        sprintf(seta2, "%c", 26);
        setfillstyle(SOLID_FILL, EGA_CYAN);
        bar(13, ymax - 2 * ymax / 40 + 1, xmax - 1, ymax - 1);
        setcolor(EGA_RED);
        outtextxy(1 * xmax / 5 - 20, ymax - (ymax / 26) / 2, seta1);
        outtextxy(1 * xmax / 5 - 10, ymax - (ymax / 26) / 2, seta2);
        outtextxy(3 * xmax / 5 - 50, ymax - (ymax / 26) / 2, "<<ENTER>>");
        setcolor(EGA_WHITE);
        outtextxy(1 * xmax / 5 + 30, ymax - (ymax / 26) / 2, "ALTERAR");
        outtextxy(3 * xmax / 5 + 45, ymax - (ymax / 26) / 2, "VALIDAR VALOR");
        break;
    case 3:
        setfillstyle(SOLID_FILL, EGA_BLUE);
        bar(13, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
        outtextxy(xmax / 4 + 30, 4 * ymax / 5 + 40, "Iniciar - para comecar aquisicao ");

        break;
    case 4:
        setfillstyle(SOLID_FILL, EGA_BLUE);

```

```

        bar(13, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
        outtextxy( xmax / 4 + 60, 4 * ymax / 5 + 40, " Qualquer tecla para parar ");
        break;

// informa como escolher o valor minimo para x
case 5:
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(13, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
    outtextxy( xmax / 4 + 60, 4 * ymax / 5 + 40, " Escolha o valor MINIMO para x");
    break;

// informa como escolher o valor maximo para x
case 6:
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(13, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
    outtextxy( xmax / 4 + 60, 4 * ymax / 5 + 40, " Escolha o valor MAXIMO para x");
    break;

case 7:
    setfillstyle(SOLID_FILL, RED);
    bar(xmax/4+180, 4 * ymax / 5 + 30, xmax/4 + 380, 16 * ymax / 20 + 40);
    outtextxy(xmax / 4 + 280, 4 * ymax / 5 + 40, "FIM D0 EXAME");
    break;

case 8:
    setcolor(WHITE);
    outtextxy(xmax / 4 + 30, 4 * ymax / 5 + 40, "A aquisicao esta parada");
    break;

case 9:
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(xmax/2, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
    setcolor(WHITE);
    outtextxy(xmax / 2 + 150/* /4 + 30*/ , 4 * ymax / 5 + 40, "A aquisicao esta parada");
    break;

case 10:
    sprintf(set1, "%c", 27);
    sprintf(set2, "%c", 26);
    setfillstyle(SOLID_FILL, EGA_CYAN);
    bar(13, ymax - 2 * ymax / 40 + 1, xmax - 1, ymax - 1);
    setcolor(EGA_RED);
    outtextxy(2 * xmax / 5/* - 10*/ , ymax - (ymax / 26) / 2, "<<QUALQUER TECLA>>");
    setcolor(EGA_WHITE);
    outtextxy(2 * xmax / 5 + 120/*55*/ , ymax - (ymax / 26) / 2, "PARAR");
    break;

case 11:
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(13, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
    outtextxy(xmax / 4 + 30, /*16*/4 * ymax / 5 + 40/*20 + 25*/ , "Porto serie (1/2):");
    setfillstyle(SOLID_FILL, EGA_CYAN);
    bar(13, ymax - 2 * ymax / 40 + 1, xmax - 1, ymax - 1);
    setcolor(EGA_RED);
    outtextxy(1 * xmax / 5 - 10, ymax - (ymax / 26) / 2, "<<1/2>>");
    setcolor(EGA_WHITE);
    outtextxy(1 * xmax / 5 + 60, ymax - (ymax / 26) / 2, "COM1/COM2");
    break;

case 12:

```

```

        setfillstyle(SOLID_FILL, EGA_CYAN);
        bar(501, 146, 638, 189/* / 2 - 1*/);
        setcolor(BLACK);
        outtextxy(570, 160, "Porto serie ");
        if (porto=='1')
            outtextxy(570, 175, "COM1");
        else
            outtextxy(570, 175, "COM2");
        break;

case 13:
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(13, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
    setcolor(RED);
    outtextxy(xmax / 4 + 30, 4 * ymax / 5 + 40, "Comunicacao nao estabelecida");
    break;

case 14:
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(xmax/2, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
    setcolor(BLACK);
    outtextxy(xmax / 2 + 150, 4 * ymax / 5 + 40, "a receber dados do sensor");
    break;

case 15:
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(xmax/2, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
    setcolor(RED);
    outtextxy(xmax / 2 + 150, 4 * ymax / 5 + 40, "Erro na comunicacao");
    break;

case 16:
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(xmax/2, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
    setcolor(RED);
    outtextxy(xmax / 2 + 150, 4 * ymax / 5 + 40, "Nao recebe dados do sensor");
    break;

case 17:
    sprintf(seta1, "%c", 27);
    sprintf(seta2, "%c", 26);
    setfillstyle(SOLID_FILL, EGA_CYAN);
    bar(13, ymax - 2 * ymax / 40 + 1, xmax - 1, ymax - 1);
    setcolor(EGA_RED);
    outtextxy(2 * xmax / 5/* - 10*/ , ymax - (ymax / 26) / 2, "<<I ou i>>");
    setcolor(EGA_WHITE);
    outtextxy(2 * xmax / 5 + 120/*55*/ , ymax - (ymax / 26) / 2, "INICIAR");

#if 0
        setfillstyle(SOLID_FILL, EGA_BLUE);
        bar(xmax/2, 4 * ymax / 5 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
        setcolor(WHITE);
        outtextxy(xmax / 2 + 150, 4 * ymax / 5 + 40, "Espera tecla de inicio (I ou i)");
#endif

        break;
    }
}
/***** fim menu_de_ajudas() *****/

```

```

/*****
/* Funcao que move o menus de seleccao */
/*****
void mover_menu() {
    int teste = 0, tecla;
    menu_de_ajudas(1); /* escreve RETURN, MOVER, CONTINUA*/
    setcolor(EGA_WHITE);
    seleccao(escolha); /* escolha = 1: escreve INICIAR*/
    while (teste != 1) {
        while( (tecla = getch()) != 0) {
            if (tecla == RETURN) { /*so para quando se carrega em RETURN*/
                teste = 1;
                break;
            }
        }
        if (tecla != RETURN) {
            tecla = getch();
            switch (tecla) {
                case 72: /* seta para cima */
                    if (escolha == 1) {
                        escolha = 5;
                        setcolor(EGA_BLACK);
                        seleccao(1);
                    }
                    else {
                        setcolor(EGA_BLACK);
                        seleccao(escolha);
                        escolha = escolha - 1;
                    }
                    setcolor(EGA_WHITE);
                    seleccao(escolha);
                    break;
                case 80: /* seta para baixo */
                    if (escolha == 7) {
                        escolha = 1;
                        setcolor(EGA_BLACK);
                        seleccao(7);
                    }
                    else {
                        setcolor(EGA_BLACK);
                        seleccao(escolha);
                        escolha = escolha + 1;
                    }
                    setcolor(EGA_WHITE);
                    seleccao(escolha);
                    break;
            }
        }
    }
}
/***** fim mover_menu() *****/

/*****
/* Funcao que abre a funcao seleccionada */
/*****
void seleccao(int a) {
    switch (a) {

```



```

    case 1:
        outtextxy(597, ymax / 20, " Iniciar ");
        break;
    case 2:
        outtextxy(597, 2 * ymax / 20, " Configurar ");
        break;
    case 3:
        outtextxy(597, 3 * ymax / 20, " Apagar ");
        break;
#if 0
    case 4:
        outtextxy(xmax - xmax / 10 + 4, 4 * ymax / 20, " Sample Rate ");
        break;
#endif
    case 4:
        outtextxy(597, 4 * ymax / 20, " Dos Shell ");
        break;
#if 0
    case 5:
        outtextxy(xmax - xmax / 10, 5 * ymax / 20, " Escala Cima");
        break;
    case 6:
        outtextxy(xmax - xmax / 10, 6 * ymax / 20, " Escala Baixo");
        break;
#endif
    case 5:
        outtextxy(597, 5 * ymax / 20, " Sair");
        break;
}
}
/***** fim selecao() *****/

void funcao_escolhida(int a) {
    switch (a) {
        case 1: ftempo_port(); break;
        case 2: configurar(); break;
        case 3: apaga(); break;
        case 4: shell(); break;
        case 5: sair = 1; break;
    }
}
/***** fim funcao_escolhida() *****/

#if 0
/*****
/* Funcao que permite alterar o valor do sample rate */
*****/
void funcao_sample_rate(){
    char buffer[37], valor[35], *valorI;
    menu_de_ajudas(2);
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(13, 3 * ymax / 4 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
    setcolor(EGA_WHITE);
    outtextxy( xmax / 4, 3 * ymax / 4 + 35, " Sample rate (em decimos de segundo) = ");
    setcolor(EGA_RED);
    outtextxy( xmax / 4, 3 * ymax / 4 + 55, " Introduza o intervalo de ");
    outtextxy( xmax / 4 + 200, 3 * ymax / 4 + 55, " tempo entre cada aquisicao");
}

```

```

do {
    setfillstyle(SOLID_FILL, EGA_BLACK);
    bar( xmax / 2 + 1, 16 * ymax / 20 , xmax / 2 + 40, 16 * ymax / 20 + 18);
    gotoxy(41, 25);
    buffer[0] = 4;
    cgets(buffer);
    strcpy(valor, &buffer[2]);
    sample_rate = strtod(valor, &valor1);
    if (sample_rate < 1 || sample_rate > 600) {
        sound(2800);
        delay(150);
        nosound();
    }
} while(sample_rate < 1 || sample_rate > 600);
setfillstyle(SOLID_FILL, EGA_BLUE);
bar(xmax - xmax / 10 - 8, 7 * ymax / 20 + 18, xmax - xmax / 10 + 27, 7 * ymax / 20 + 36);
/*sprintf(valor, "%3d", sample_rate);
outtextxy(xmax - xmax / 10 + 20, 7 * ymax / 20 + 60, valor);*/
/*sample_rate = sample_rate/5;*/
menu_de_ajudas(1);
/*settimer;*/
    apaga();
    sprintf(valor, "%d", sample_rate);
    outtextxy(xmax - 50, 7 * ymax / 20 + 8 + 20 + 60, valor);
    setfillstyle(SOLID_FILL, EGA_BLUE);
    bar(13, 3 * ymax / 4 + 20, xmax - 1, ymax - 2 * ymax / 40 - 7);
    outtextxy(xmax / 4 + 30, 16 * ymax / 20 + 15, "A aquisiç, o est parada");
}
/***** fim funcao_sample_rate() *****/
#endif

/*****
/* Funcao que permite ampliar o eixo dos xx do grafico das medias (cima) */
/*****
void escala_cima() {
    menu_de_ajudas(2);
    menu_de_ajudas(5);
    while (sair == 0) { // espera por uma tecla e
        while( (tecla = getch()) != 0) { // repete ate tecla ser uma
            if (tecla == RETURN) { // das setas horizontais ou RETURN
                sair = 1; // se tecla for RETURN sai
                break;
            }
        }
    }
    if (tecla != RETURN) { // se tecla for uma das setas horizontais
        tecla = getch();
        switch (tecla) {
            case LEFT: // seta para esquerda
                if (ampli_min_cima > 0) { // se valor minimo de x e maior que 0
                    min_ant_cima = ampli_min_cima;
                    max_ant_cima = ampli_max_cima;
                    ampli_min_cima = ampli_min_cima - 20; // novo valor minimo de x
                    coord_xx_cima(ampli_max_cima, ampli_min_cima); // calcula a
nova escala
                    graf_ampli_cima(); // imprime o grafico ampliado
                }
                break;

```

```

        case RIGHT:                // seta para direita
            if (ampli_min_cima < ampli_max_cima - 20) { // se o valor minimo de x for
menor que (valor maximo - 20)
                min_ant_cima = ampli_min_cima;
                max_ant_cima = ampli_max_cima;
                ampli_min_cima = ampli_min_cima + 20; // novo valor minimo de x
                coord_xx_cima(ampli_max_cima,ampli_min_cima); // calcula a nova escala
                graf_ampli_cima(); // imprime o grafico ampliado
            }
            break;
        }
    }
}
sair = 0;
menu_de_ajudas(6);

while (sair == 0) { // espera por uma tecla e
    while( (tecla = getch()) != 0) { // repete ate tecla ser uma
        if (tecla == RETURN) { // das setas horizontais ou RETURN
            sair = 1; // se tecla for RETURN sai
            break;
        }
    }
    if (tecla != RETURN) { // se tecla for uma das setas horizontais
        tecla = getch();
        switch (tecla) {
            case LEFT:                // seta para esquerda
                if (ampli_max_cima > ampli_min_cima + 20) { // se o valor maximo de x
                min_ant_cima = ampli_min_cima; // for maior que (valor minimo + 20)
                max_ant_cima = ampli_max_cima;
                ampli_max_cima = ampli_max_cima - 20; // novo valor maximo para x
                coord_xx_cima(ampli_max_cima,ampli_min_cima); // calcula a nova escala
                graf_ampli_cima(); // imprime o grafico ampliado
            }
            break;
            case RIGHT:                // seta para direita
                if (ampli_max_cima < NDAT) { // se o valor maximo de x for menor
                min_ant_cima = ampli_min_cima; // que o numero maximo de canais
                max_ant_cima = ampli_max_cima;
                ampli_max_cima = ampli_max_cima + 20; // nono valor maximo de x
                coord_xx_cima(ampli_max_cima,ampli_min_cima); // calcula a nova escala
                graf_ampli_cima(); // imprime o grafico ampliado
            }
            break;
        }
    }
}
sair = 0;
menu_de_ajudas(3);
}
/***** fim escalas() *****/

/*****
/* Funcao que permite ampliar o eixo dos xx do grafico dos valores */
/* do ADC (baixo) */
*****/
void escala_baixo() {

```

```

menu_de_ajudas(2);
menu_de_ajudas(5);
while (sair == 0) { // espera por uma tecla e
while( (tecla = getch()) != 0) { // repete ate tecla ser uma
    if (tecla == RETURN) { // das setas horizontais ou RETURN
        sair = 1; // se tecla for RETURN sai
        break;
    }
}
if (tecla != RETURN) { // se tecla for uma das setas horizontais
    tecla = getch();
    switch (tecla) {
        case LEFT: // seta para esquerda
            if (ampli_min_baixo > 0) { // se valor minimo de x e maior que 0
                min_ant_baixo = ampli_min_baixo;
                max_ant_baixo = ampli_max_baixo;
                ampli_min_baixo = ampli_min_baixo - 20; // novo valor minimo de x
                coord_xx_baixo(ampli_max_baixo,ampli_min_baixo); // calcula a nova escala
                graf_ampli_baixo(); // imprime o grafico ampliado
            }
            break;
        case RIGHT: // seta para direita
            if (ampli_min_baixo < ampli_max_baixo - 20) { // se o valor minimo de x for
menor que (valor maximo - 20)
                min_ant_baixo = ampli_min_baixo;
                max_ant_baixo = ampli_max_baixo;
                ampli_min_baixo = ampli_min_baixo + 20; // novo valor minimo de x
                coord_xx_baixo(ampli_max_baixo,ampli_min_baixo); // calcula a
nova escala
                graf_ampli_baixo(); // imprime o grafico ampliado
            }
            break;
    }
}
}
sair = 0;
menu_de_ajudas(6);
while (sair == 0) { // espera por uma tecla e
while( (tecla = getch()) != 0) { // repete ate tecla ser uma
    if (tecla == RETURN) { // das setas horizontais ou RETURN
        sair = 1; // se tecla for RETURN sai
        break;
    }
}
if (tecla != RETURN) { // se tecla for uma das setas horizontais
    tecla = getch();
    switch (tecla) {
        case LEFT: // seta para esquerda
            if (ampli_max_baixo > ampli_min_baixo + 20) { // se o valor maximo de x
                min_ant_baixo = ampli_min_baixo; // for maior que (valor minimo + 20)
                max_ant_baixo = ampli_max_baixo;
                ampli_max_baixo = ampli_max_baixo - 20; // novo valor maximo para x
                coord_xx_baixo(ampli_max_baixo,ampli_min_baixo); // calcula a
nova escala
                graf_ampli_baixo(); // imprime o grafico ampliado
            }

```

```

        break;
    case RIGHT:          // seta para direita
        if (ampli_max_baixo < 500) { // se o valor maximo de x for menor
            min_ant_baixo = ampli_min_baixo; // que o numero maximo de canais
            max_ant_baixo = ampli_max_baixo;
            ampli_max_baixo = ampli_max_baixo + 20; // novo valor maximo de x
            coord_xx_baixo(ampli_max_baixo,ampli_min_baixo); // calcula a
nova escala
            graf_ampli_baixo(); // imprime o grafico ampliado
        }
        break;
    }
}
}
sair = 0;
menu_de_ajudas(3);
}
/***** fim escalas() *****/

/*****
/* Funcao que permite visualizar o grafico das medias (cima) */
/* com a nova escala, pintando os pixel correspondentes */
*****/
void graf_ampli_cima() {
    int valor_x1, valor_y1;

    for (i = min_ant_cima; i < max_ant_cima; i++) {
        valor_x1 = ceil(13 + dx_cima * (i - min_ant_cima));
        valor_y1 = ((bufadc[i] >> 4) & 0xff);
        putpixel(valor_x1, 140 - valor_y1, BLACK);
    }
    dx_cima = NDAT / (ampli_max_cima - ampli_min_cima); // novo factor de escala do eixo xx
    for (i = ampli_min_cima; i < ampli_max_cima; i++) {
        valor_x1 = ceil(13 + dx_cima * (i - ampli_min_cima)); // imprime no ecrã
        valor_y1 = ((bufadc[i] >> 4) & 0xff);
        putpixel(valor_x1, 140 - valor_y1, CYAN);
    }

    grafico();
    apresentacao();
}
/***** fim graf_ampli_cima() *****/

/*****
/* Funcao que permite visualizar o grafico dos valores do ADC (baixo) */
/* com a nova escala, pintando os pixel correspondentes */
*****/
void graf_ampli_baixo() {
    int valor_x1, valor_y1;
    for (i = min_ant_baixo; i < max_ant_baixo; i++) {
        valor_x1 = ceil(13 + dx_baixo * (i - min_ant_baixo));
        valor_y1 = ((bufadc[i] >> 4) & 0xff);
        putpixel(valor_x1, 328 - valor_y1, BLACK);
    }
    dx_baixo = NDAT / (ampli_max_baixo - ampli_min_baixo); // novo factor de escala do eixo xx

    for (i = ampli_min_baixo; i < ampli_max_baixo; i++) {
        valor_x1 = ceil(13 + dx_baixo * (i - ampli_min_baixo)); // imprime no ecrã

```

```

    valor_y1 = ((bufadc[i] >> 4) & 0xff);
    putpixel(valor_x1, 328 - valor_y1, CYAN);
}

grafico();
apresentacao();
}
/***** fim graf_ampli_baixo() *****/

void coord_xx_cima(float ampli_max, float ampli_min) {
    char coord[24];

    setfillstyle(SOLID_FILL, EGA_BLACK);
    bar(0, 2 * ymax / 5 - 8, xmax - xmax / 5 - 50, 2 * ymax / 5 + 2);

    settxtjustify(CENTER_TEXT, TOP_TEXT);
    settxtstyle(SMALL_FONT, HORIZ_DIR, 4);

    sprintf(coord, "%2.0f", (ampli_min + 7 * (ampli_max - ampli_min) / 7));
    outtextxy(7 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 6, coord);

    sprintf(coord, "%2.0f", (ampli_min + 6 * (ampli_max - ampli_min) / 7));
    outtextxy(6 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 6, coord);

    sprintf(coord, "%2.0f", (ampli_min + 5 * (ampli_max - ampli_min) / 7));
    outtextxy(5 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 6, coord);

    sprintf(coord, "%2.0f", (ampli_min + 4 * (ampli_max - ampli_min) / 7));
    outtextxy(4 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 6, coord);

    sprintf(coord, "%2.0f", (ampli_min + 3 * (ampli_max - ampli_min) / 7));
    outtextxy(3 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 6, coord);

    sprintf(coord, "%1.0f", (ampli_min + 2 * (ampli_max - ampli_min) / 7));
    outtextxy(2 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 6, coord);

    sprintf(coord, "%1.0f", (ampli_min + 1 * (ampli_max - ampli_min) / 7));
    outtextxy(1 * (7 * 64) / 7 + 12, 2 * ymax / 5 - 6, coord);

    sprintf(coord, "%1.0f", (ampli_min + 0 * (ampli_max - ampli_min) / 7));
    outtextxy(12, 2 * ymax / 5 - 6, coord);
    settxtjustify(CENTER_TEXT, CENTER_TEXT);
    settxtstyle(DEFAULT_FONT, HORIZ_DIR, 1);
}

/***** fim coord_x_cima() *****/

void coord_xx_baixo(float ampli_max, float ampli_min) {
    char coord[24];
    setfillstyle(SOLID_FILL, EGA_BLACK);
    bar(0, 4 * ymax / 5 + 2, xmax - 1, 4 * ymax / 5 + 12);
    settxtjustify(CENTER_TEXT, TOP_TEXT);
    settxtstyle(SMALL_FONT, HORIZ_DIR, 4);

    /* Escala horizontal */
    sprintf(coord, "%2.0f", (ampli_min + 9 * (ampli_max - ampli_min) / 9));
    outtextxy(525, 380, coord);
    sprintf(coord, "%2.0f", (ampli_min + 8 * (ampli_max - ampli_min) / 9));
    outtextxy(8 * 513 / 9 + 12, 380, coord);
    sprintf(coord, "%2.0f", (ampli_min + 7 * (ampli_max - ampli_min) / 9));
    outtextxy(7 * 513 / 9 + 12, 380, coord);
    sprintf(coord, "%2.0f", (ampli_min + 6 * (ampli_max - ampli_min) / 9));

```

```

    outtextxy(6 * 513 / 9 + 12 , 380, coord);
    sprintf(coord, "%2.0f", (ampli_min + 5 * (ampli_max - ampli_min) / 9));
    outtextxy(5 * 513 / 9 + 12 , 380, coord);
    sprintf(coord, "%1.0f", (ampli_min + 4 * (ampli_max - ampli_min) / 9));
    outtextxy(4 * 513 / 9 + 12 , 380, coord);
    sprintf(coord, "%1.0f", (ampli_min + 3 * (ampli_max - ampli_min) / 9));
    outtextxy(3 * 513 / 9 + 12 , 380, coord);
    sprintf(coord, "%1.0f", (ampli_min + 2 * (ampli_max - ampli_min) / 9));
    outtextxy(2 * 513 / 9 + 12 , 380, coord);
    sprintf(coord, "%1.0f", (ampli_min + 1 * (ampli_max - ampli_min) / 9));
    outtextxy(1 * 513 / 9 + 12 , 380, coord);
    sprintf(coord, "%1.0f", (ampli_min + 0 * (ampli_max - ampli_min) / 9));
    outtextxy(12 , 380, coord);

/* Escala vertical */
    sprintf(coord, "%d", 300);
    outtextxy(540, 347, coord);
    sprintf(coord, "%d", 400);
    outtextxy(540, 297, coord);
    sprintf(coord, "%d", 500);
    outtextxy(540 , 247, coord);
    sprintf(coord, "%d", 600);
    outtextxy(540 , 197, coord);
    sprintf(coord, "%d", 700);
    outtextxy(540 , 147, coord);
    sprintf(coord, "%d", 800);
    outtextxy(540 , 97, coord);
    sprintf(coord, "%d", 900);
    outtextxy(540 , 47, coord);
    settxtjustify(CENTER_TEXT, CENTER_TEXT);
    settxtstyle(DEFAULT_FONT, HORIZ_DIR, 1);
}
/***** fim coord_x *****/

void main()
{
    opengraph();
    clearviewport();
    xmax = getmaxx();
    ymax = getmaxy();
    apresentacao1(); /*desenha a janela inicial*/
    clearviewport();
    setbkcolor(EGA_BLACK);
    settxtjustify(CENTER_TEXT, CENTER_TEXT);
    apresentacao(); /*desenha o menu principal*/
    sair = 0;
    conf = 0x0;
    escolha = 1;
    porto = '2'; /*COM2 por defeito*/
    menu_de_ajudas(3); /*escreve:"iniciar - para comecar aquisicao"*/
    do {
        mover_menu();
        funcao_escolhida(escolha);
    } while (sair == 0);
    closegraph();
}

```