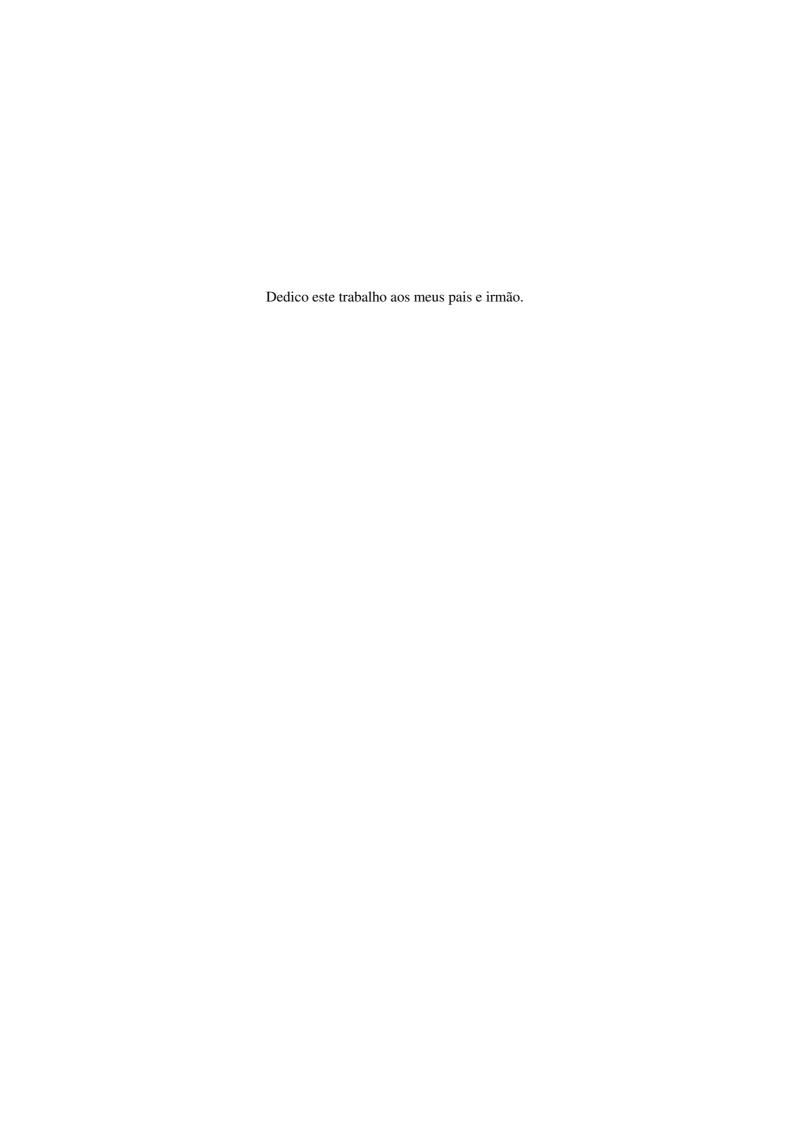
Pedro Miguel Chambel Garcia Gomes Henriques INTERLIGAÇÃO DE SENSORES ATRAVÉS DO CORPO HUMANO: CARACTERIZAÇÃO E MODELAÇÃO

Telecomunicações e Informática

Pedro Miguel Chambel Garcia Gomes Henriques INTERLIGAÇÃO DE SENSORES ATRAVÉS DO CORPO HUMANO: CARACTERIZAÇÃO E **MODELAÇÃO**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Electrónica e Telecomunicações, realizada sob a orientação científica do Prof. Doutor Rui Manuel Ramos Escadas Martins, professor do Departamento de Electrónica e Telecomunicações e Informática da Universidade de Aveiro.



O júri

Presidente

Prof. Dr. António Luís Jesus Teixeira Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Vogais

Prof. Dr. Rui Manuel Escadas Ramos Martins Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro. (Orientador)

Prof. Dr. José Miguel Costa Dias Pereira Professor Coordenador do Departamento de Sistemas e Informática da Escola Superior de Tecnologia de Setúbal

Agradecimentos

Gostava de aproveitar este meio para agradecer o meu professor orientador Prof. Rui Manuel Escadas Martins por toda a motivação que me deu e pela confiança que me depositou, mas sobretudo ao apoio prestado durante a realização deste trabalho.

Gostaria também de expressar o meu agradecimento aos meus pais e irmão que me apoiaram incondicionalmente durante todo o percurso académico, a todos os meus colegas de curso, aos meus colegas do laboratório 234 e um agradecimento especial ao João Lima, Hélder Moura e Cesário Lucas, as pessoas com quem mais convivi durante os últimos 5 anos e que estiveram sempre presentes para me ajudar e aconselhar.

Não podia deixar de agradecer a ajuda prestada pelo técnico do Instituto de Telecomunicações Paulo Gonçalves que me ajudou na assemblagem de alguns componentes e ao Sr. Pereira e Sr. Simões pela ajuda prestada.

Finalmente, queria também prestar o meu agradecimento aos meus colegas que residem comigo pela toda ajuda, compressão e apoio durante o decorrer deste trabalho.

Palavras-chave

DDS, micro controladores, Body Area Network, Vector analyser, Human body comunication

Resumo

O projecto teve como objectivo a criação de um setup que pudesse ser portátil e que permitisse comparar dois sinais sinusoidais com uma frequência até aos 166MHz e que indica-se qual a razão de amplitude e a diferença de fase entre os dois sinais.

Este sistema foi concebido com o intuito de efectuar uma caracterização de sinais que atravessam pelo corpo humano determinando as melhores condições para a transmissão destes, ou seja, qual a(s) frequência que apresenta a melhor resposta.

A caracterização de sinais que atravessam pelo corpo humano é enquadrada no desenvolvimento e estudo de redes BAN's em que se pretende efectuar a transmissão de sinais pelo corpo humano de forma a colmatar as desvantagens de outras tecnologias até á data existentes, mais em concreto o consumo energético.

O sistema é baseado num gerador de frequências programável, DDS, onde o sinal gerado é acondicionado através de amplificadores/ atenuadores, tudo controlado digitalmente através de um micro controlador. O sinal emitido tem como objectivo excitar o dispositivo em teste. O sinal proveniente do dispositivo em teste é comparado com o emitido através do uso de detectores de sinais. Estes produzem dois sinais DC com amplitude em função da razão entre amplitude e diferença de fase.

Relativamente à interface com o utilizador, foi elaborada uma interface gráfica com o intuito de permitir uma abstracção do sistema construído por parte do utilizador, permitindo também uma melhor visualização dos resultados obtidos.

Keywords

DDS, Microcontrollers, Body Area Network, Vector analyser, Human body comunication

Abstract

The objective of this project was the creation of a Setup that could be portable and at the same time allowed the user to compare a sine wave with a top frequency of 166MHz. This Setup should also be able to indicate the ratio between the amplitude and the wave difference of two difference signals.

This system was developed with the notion of being able to characterize signals that run through the human body, determining the idle conditions for the transmission of these same signals. In others words, the system looks for the frequency that has a better response.

The characterization of signals that travel through the human body is nested in the study and development of BAN networks where the main objective is to transmit signals through the body in order to fill in the gap the exists of other existing technologies such as energetic consumption.

The system developed is based on a programmable frequency generator. The signal that is created with the generator is fed through amplifiers and attenuators and is digitally controlled with a microcontroller. The signal that is emitted will then cross a Device Under Test (DUT). The signal that comes out the DUT will then be compared with the one that was initially emitted from the generator. This will be done with the use of signal detectors. The signal detectors will then produce two distinct DC signals with specific amplitude. This amplitude represents the ratio between the amplitude and the phase difference of the initial entering signals.

For the user interface, a graphic interface was developed bearing in mind an abstraction of the entire system. This simplifies the task of the user and at the same time allows a better understanding and comprehension of the final results.

Conteúdo

1	Intr	oduçã	o	1
	1.1	Motiv	ação	1
	1.2	Objec	tivos	2
2	Вос	ly Are	a Network	5
	2.1	Introd	lução	5
	2.2	Body	area network vs outras tecnologias	ŝ
		2.2.1	Bluetooh	ŝ
		2.2.2	Wi-Fi	7
		2.2.3	ZigBee	3
		2.2.4	Resumo	9
	2.3	Aplica	ıções de BANs)
		2.3.1	Monitorização da actividade física)
		2.3.2	Identificação e segurança	1
		2.3.3	Transmissão de sinais de áudio	1
		2.3.4	Rede de sensores do corpo humano	2
		2.3.5	Situações de emergência	3
		2.3.6	Aplicações Militares	4
	2.4	Projec	ctos BAN reportados na literatura	4
		2.4.1	RedTacton	5
		2.4.2	ETRI	5
		2.4.3	Transceivers existentes	3
			2.4.3.1 Zimmerman	3
			2.4.3.2 Hachisuka	7
			2.4.3.3 Shinagawa	3
			2.4.3.4 Song	3
			2.4.3.5 Resumo	9
	2.5	Propa	gação de sinais através do corpo humano	a

ii $CONTE\'{U}DO$

	2.6	DDS	
		2.6.1	Arquitectura
		2.6.2	Jitter
		2.6.3	SPI
	2.7	Netwo	$rk \ analyzer \ \dots \ \dots \ \dots \ 30$
		2.7.1	Parâmetros S
		2.7.2	Varrimento na frequência
3	Har	rdware	35
	3.1	Mothe	rboard
		3.1.1	Microcontrolador
		3.1.2	DDS
		3.1.3	<i>DDS</i> AD9859
		3.1.4	Filtro <i>DDS</i>
		3.1.5	Atenuador
		3.1.6	Detector de sinal
		3.1.7	Funcionamento
		3.1.8	Aplicações
			3.1.8.1 Filtro Detector
		3.1.9	Selecção do parâmetro a analisar
		3.1.10	Circuito Fonte de sinal
		3.1.11	LCD
	3.2	Pré Ai	mplificador de entrada
	3.3	Placa	de Alimentação
		3.3.1	Tensão de +5V e -5V
			3.3.1.1 Alimentação através da rede
			3.3.1.2 Alimentação através do USB
		3.3.2	Tensão de 3.3V, 3V, 3.069 e 1.8V
		3.3.3	Sensor de temperatura e FAN
4	Sof	tware	57
	4.1	Interfa	ce Gráfica
		4.1.1	Tramas de envio
	4.2	Micro	controlador
		4.2.1	Calibração
		4.2.2	Cálculo das frequências
		4.2.3	Geração de frequências

iii

		4.2.4	Determinação dos parâmetros S	. 67
5	Res	ultado	S	69
	5.1	Pcb's		. 69
	5.2	Forma	s de onda do sinal gerado	. 71
		5.2.1	Formas de onda à saída do DDS	. 71
		5.2.2	Formas de onda à saída do Opamp configuração diferença	. 72
		5.2.3	Formas de onda à saída do Filtro fc=166MHz $\dots \dots$. 73
		5.2.4	Formas de onda à entrada do detector de sinal	. 73
	5.3	Respos	sta do detector de sinal	. 74
	5.4	Calibra	ação	. 75
		5.4.1	Amplitude	. 75
		5.4.2	Fase	. 76
6	Con	clusõe	s e Trabalho Futuro	77
\mathbf{A}	Cód	ligo PI	\mathbf{C}	83
	A.1	Menu.	c	. 84
	A.2	Driver	.c	. 87
	A.3	Driver	.h	. 108
В	Cód	ligo Vi	sual Basic	111
	B.1	Form1	.vb	. 112
	B.2	Grafice	ocalibracao.vb	. 123
	В.3	Grafice	o.vb	. 124
\mathbf{C}	Esq	uemáti	ico Motherboard	129
D	Esq	uemáti	ico Placa pré amplificador de entrada	135
${f E}$	Esq	uemáti	ico Filtros	137
\mathbf{F}	Esq	uemáti	ico LCD	139
\mathbf{G}	Esq	uemáti	ico Placa de Alimentação	141
н	Har	dware	total	145

Lista de Figuras

1.1	Situação onde as referências se encontram curto-circuitadas	2
1.2	Corpo humano como sistema capacitivo	2
2.1	Rede BAN invasivo [1]	6
2.2	Comunicação Bluetooh [2]	7
2.3	Comunicação Wi-Fi [3]	8
2.4	Módulo ZigBee [4]	9
2.5	Relação consumo/taxa de transferência por tecnologia [5]	10
2.6	Atleta com mp3 que visualiza a actividade corporal [5]	11
2.7	Implementação de sistema de identificação [6]	11
2.8	Transmissão de sinal de áudio [5]	12
2.9	Monitorização de uma rede de sensores [5]	13
2.10	Diagrama de comunicação entre um paciente e o serviço de urgência [7]	13
2.11	Bombeiro equipado com sistema BAN [8]	14
2.12	Diagrama de blocos do modo de funcionamento [6]	15
2.13	Protótipo RedTacton desenvolido [6]	15
2.14	Protótipo ETRI [9]	16
2.15	Principio de funcionamento do protótipo ETRI [9]	16
2.16	Diagrama de blocos Zimmerman [10]	17
2.17	Protótipo Zimmerman desenvolvido [10]	17
2.18	Protótipo Hachisuka [11]	18
2.19	Configuração utilizada [12]	20
2.20	Modelos de medida [12]	20
2.21	Resposta em frequência quando o individuo se encontra sentado	21
2.22	Resposta em frequência quando o individuo se encontra de pé . . .	21
2.23	Resposta EVM em função de vários factores [12]	22
2.24	Configuração utilizada [11]	23
2.25	Resposta em frequência na comunicação pelo corpo humano e pelo ar [11]	23

2.26	Configuração utilizada para a comunicação
2.27	Diagrama de transferência de pacotes
2.28	Diagrama de blocos simplificado de um DDS [13]
2.29	Roda ilustrativa [13]
2.30	Vários passos para criação de um sinal [13]
2.31	DDS sincronizados para modulação I-Q [14]
2.32	Redução da influência do jitter no sinal de saída do DDS [15] 2
2.33	Comunicação SPI entre Master/Slave [16]
2.34	Lista de dispositivos possíveis de analisar com o network analyzer [17] . 3
2.35	Diagrama de blocos de um vector network analyzer [17]
2.36	Diagrama de reflexões
2.37	Variação de frequência entre o sinal emitido e recebido [18]
3.1	Diagrama de blocos da motherboard
3.2	Configuração utilizada para o PIC
3.3	Configuração DDS
3.4	Diagrama Funcional do DDS AD9859 [14]
3.5	Opamp na configuração diferença
3.6	Filtro projectado
3.7	Resposta do filtro
3.8	Análise de Monte Carlo do filtro
3.9	Atenuador de 19.45dBV
3.10	Diagrama de blocos AD8302 [19]
3.11	Resposta com a variação de amplitude e diferença de fase [19] 4
3.12	Dispositivo operando no modo de controlador [19]
3.13	Configuração do detector de sinal
3.14	Filtros que o sinal atravessa em função da frequência do sinal RF $$ $$ 4
3.15	Configuração do filtro de 100Hz
3.16	Configuração do filtro de 3.3KHz
3.17	Configuração do filtro de 100KHz
3.18	configuração do filtro de 3.3 MHz
3.19	Análise de Monte Carlo do filtro de 100Hz
3.20	Análise de Monte Carlo do filtro de 3.3KHz
3.21	Análise de Monte Carlo do filtro de 100KHz
3.22	Análise de Monte Carlo do filtro de 3.3MHz
3.23	Configuração utilizada para o Multiplex
3.24	Configuração utilizada para o 74HC4052

LISTA DE FIGURAS	vii

3.25	Configuração do amplificador	48
3.26	Configuração do Relay	49
3.27	Configuração atenuador digital	50
3.28	Diagrama temporal de transferência para o atenuador digital	50
3.29	Configuração utilizada para o LCD	51
3.30	Diagrama de blocos da placa pré amplificador de entrada	51
3.31	Diagrama de blocos da placa de alimentação	53
3.32	Configuração utilizada para alimentação através da rede	54
3.33	Configuração utilizada para para a alimentação através do USB $$	54
3.34	Configuração utilizada para obter 3.3V, 3V, 3.069V e 1.8V $$	55
3.35	Configuração utilizada para o sensor de temperatura e para a ventoinha	56
4.1	Diagrama geral de código do microcontrolador	57
4.2	menu inicial da interface gráfica	58
4.3	Gráfico com a resposta em frequência da amplitude do parâmetro S_{21} .	59
4.4	Menu para guardar dados	59
4.5	Fluxograma temporal da comunicação entre o o PC e o PIC	60
4.6	Trama de envio para analisar várias frequências	60
4.7	Trama de envio para analisar uma única frequência	60
4.8	Trama enviada pelo PIC	61
4.9	Fluxograma Temporal	61
4.10	diagrama de blocos do código da calibração	63
4.11	Diagrama de blocos do código para calcular as frequências na escala	
	logaritmica	65
4.12	Diagrama de blocos do código para calcular as frequências na escala linear	66
4.13	Diagrama de blocos da geração de uma dada frequência	67
4.14	Diagrama de blocos do código para a determinação dos parâmetros S $$.	68
5.1	Pcb da placa pré amplificador de entrada	69
5.2	Pcb da placa Motherboard	70
5.3	Pcb para conexão do LCD	70
5.4	Pcb da Placa de Alimentação	71
5.5	Forma de onda com frequência de 1KHz à saída do DDS	72
5.6	Forma de onda com frequência de 1MHz à saída do DDS	72
5.7	Forma de onda com frequência de 10MHz à saída do DDS	72
5.8	Forma de onda com frequência de 1MHz à saída do opamp	73
5.9	Forma de onda com frequência de 1MHz à saída do filtro fc= $166\mathrm{MHz}$.	73

viii LISTA DE FIGURAS

5.10	Forma de onda com frequência de 1MHz à entrada do detector de sinal	74
5.11	Gráfico da tensão à saída do filtro em função da atenuação do sinal de	
	$entrada . \ . \ . \ . \ . \ . \ . \ . \ . \ .$	74
5.12	Resultados Obtidos	75

Lista de Tabelas

2.1	Comparação dos vários transceivers desenvolvidos [20]	19
2.2	Principais parâmetros [12]	20
2.3	Principais parâmetros [12]	21
2.4	Taxa de erro de pacotes [21]	24
2.5	Modos SPI existentes [16]	29
3.1	Composição do primeiro $byte$ de configuração do DDS AD9859 [14]	38

Lista de Acrónimos

BAN - Body Area Network - rede de comunicação pessoal

 DUT - Device Under Test - dispositivo em teste

ECG - eletrocardiograma

EEG - eletroencefalograma

EOF - End Off File

WBS - Wideband Sinal - sinais de banda larga

VCO - Voltage Controlled Oscillator

 ${\tt DDS} \textit{-} \textit{Direct Digital Synthesizer}$

CDR - Clock Sata Recovery

QSPD - Quadratic Sampling Phase Detector

PRBS - Pseudo-Random Binary Sequence

EVM - Error Vector Measurement

fc - Frequência de corte

Capítulo 1

Introdução

1.1 Motivação

Com a evolução e aparecimento de novas tecnologias é cada vez mais fácil interligar vários dispositivos em redes de comunicação pessoais (BAN's). Contudo as soluções existentes têm pouca imunidade às interferências, tem um alcance pouco definido o que causa problemas de segurança e apresentam baixas taxas de transmissão.

Um dos factores que se encontra em foco na actualidade é a eficiência energética. Relativamente a este ponto a aplicação da transmissão de sinais através do corpo humano é uma mais valia, dado que se consegue obter uma maior eficiência energética comparativamente com outras tecnologias como, ZigBee, Bluetooh, IEEE 803.11, Wi-Fi, MiWi, uma vez que grande parte da energia é dispersa obtendo um baixo rendimento.

De forma a contornar estes problemas o corpo humano pode ser usado como meio de comunicação. Contudo estas soluções também apresentam algumas restrições à transmissão de sinais, tais como:

- variação na resposta em frequência do corpo humano;
- variação da impedância ao longo do corpo;
- limitações legais relacionadas com a saúde do utilizador.

Sobre a propagação de sinais RF na proximidade do corpo humano existe muita informação disponível, contudo esta baseia-se essencialmente no estudo do far-field.

O que se pretende no projecto em que origina este trabalho é uma solução diferente, no qual ainda não existe um grande desenvolvimento, ou seja, pretende-se efectuar uma caracterização e modelação do meio de comunicação do corpo humano com alguma precisão e sem a perturbação criada pelos instrumentos convencionais. A solução pretendida baseia-se num sistema portátil de fácil transporte que caracterize o sistema em teste. Este sistema possui um isolamento óptico de forma a que não exista

nenhuma componente metálica que provoque campos, alterando as condições de teste.

Os grandes incentivadores do desenvolvimento e pesquisa sobre esta tecnologia são as aplicações inerentes, ou seja, existe uma vasta área de aplicações, tais como:

- observação de sinais biomédicos em tempo real;
- conexão entre vários dispositivos como PDA's, telemóveis, MP3, no qual pode permitir a simples transmissão de sinal de áudio através do corpo humano;
 - utilização para sistema de identificação pessoal.

Nada melhor do que trabalhar uma tecnologia que ainda está a dar os primeiros passos, mas com um enorme potencial, no qual pode permitir uma melhor qualidade de vida das pessoas.

1.2 Objectivos

Na medição de sinais na superfície do corpo humano é recorrentemente utilizado dois eléctrodos: o de sinal e o de referência quer para o gerador quer para o aparelho de medida. Suspeita-se que os resultados obtidos desta forma não sejam os mais correctos dado que ambas as referências se encontram curto-circuitadas, figura 1.1, mas também devido ao facto corpo humano ser um sistema capacitivo, figura 1.2.

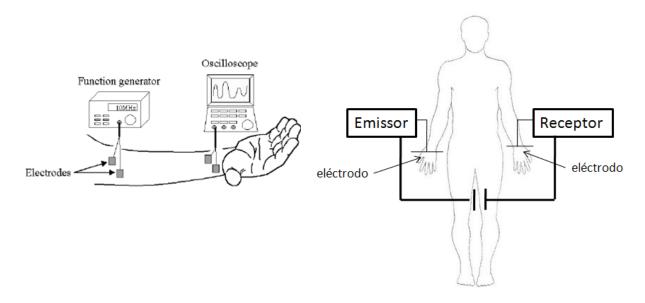


Figura 1.1: Situação onde as referências se Figura 1.2: Corpo humano como sistema encontram curto-circuitadas capacitivo [12]

Perante esta situação, propôs-se elaborar um sistema em que a medição não perturbasse nem interferisse no meio de comunicação. 1.2. OBJECTIVOS 3

O objectivo deste projecto tem como finalidade a construção de um setup que possibilite a caracterização de sinais transmitidos pelo corpo humano determinando as melhores condições para a sua transmissão, efectuando um varrimento na frequência desde os 100KHz até os 166MHz, ou seja, observar a resposta em fase e em amplitude determinando qual a(s) frequência(s) que apresenta a melhor resposta. Os resultados obtidos deverão ser visualizados numa interface user-friendly. Para este efeito vai ser projectado e construído um dispositivo semelhante a um network analyzer (com algumas limitações, nomeadamente a gama dinâmica e o número de canais) mas que seja imune às perturbações que os aparelhos convencionais deste tipo causam neste particular tipo de comunicações.

A tese está organizada da seguinte forma. O capítulo 2 contém uma descrição sobre o Body Area Network, mais propriamente as caracrterísticas que o definem, uma comparação com outras tecnologias que poderiam efectuar uma função semelhante, uma descrição das aplicações mais importantes das redes BAN, uma análise de alguns projectos reportados na literatura, uma abordagem de alguns estudos efectuados sobre a propagação de sinais através do corpo humano e por fim uma descrição dos componentes essencias e determinantes na realização deste trabalho. No capítulo 3 é apresentado todo o hardware desenvolvido e no capítulo 4 é apresentado todo o software desenvolvido, desde o código para a interface gráfica quer para o microcontrolador. No capítulo 5 encontram-se expostos os resultados obtidos e por fim no capítulo 6 são referidas as conclusões e trabalho futuro.

Capítulo 2

Body Area Network

2.1 Introdução

Os sistemas BAN foram criados com o intuito de promover a comunicação na proximidade do corpo humano. Estes sistemas deverão obedecer a alguns requisitos, tais como:

- possuir um alcance previsível entre 2 a 5m;
- conseguir interligar até 100 sensores;
- permitir uma taxa de transmissão de 1Mbit/s;
- deverá ter um consumo na ordem de 1mW/Mbit/s (@1m);
- os transceivers deverão ter uma dimensão reduzida;
- o canal de comunicação ser bidireccional [5].

As redes de comunicação BAN podem subdividir-se em 2 tipos:

- Invasivo - BANs que suportam a comunicação bilateral entre dispositivos implantados no corpo humano e estações externas, como por exemplo, um implante de um pacemaker no corpo de um indivíduo que envia periodicamente dados relativamente ao estado vital para um dispositivo, onde um médico poderá visualizar os dados recolhidos e caso seja necessário o implante poderá ser reprogramado e/ou ajustado, figura 2.1 [1].

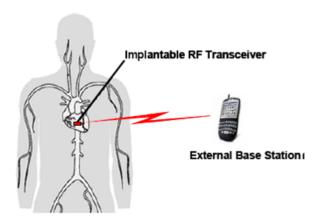


Figura 2.1: Rede BAN invasivo [1]

- Não invasivos - BANs que suportam a comunicação com dispositivos que se encontram à superfície ou perto do corpo humano e a redes de comunicação que se situem perto deste [1].

2.2 Body area network vs outras tecnologias

Como é evidente há outras tecnologias que podem realizar uma função semelhante, contudo estas apresentam algumas desvantagens que se pretendem colmatar com a transmissão através do corpo humano. De seguida segue-se uma descrição de algumas dessas tecnologias.

2.2.1 Bluetooh

Bluetooth é um protocolo padrão de comunicação sem fios desenvolvido com o objectivo de permitir a transferência de dados com um baixo consumo para um pequeno alcance. Esta tecnologia teve como fins iniciais a interligações de vários periféricos e computadores sem estarem na linha de visão uns com os outros, colmatando a desvantagem da tecnologia existente anteriormente, baseado em infravermelhos, Irda.

Principais características:

- Alcance até 100m;
- Consumo de 1mW (1m de alcance) até 100mW (100m de alcance);
- Opera na faixa de ISM, nos 2,45GHz;
- Taxa de transferência até 723,1 Kbit/s.
- Consegue operar em ambientes com elevado ruído.

Principais aplicações:

- Comunicação sem fios entre computadores, periféricos, gadjects e telemóveis;
- Utilização em consolas e equipamentos de diversão;
- Uso em equipamentos de áudio e vídeo, como leitores de mp3 e mp4;
- Utilização em impressoras e faxes.



Figura 2.2: Comunicação Bluetooh [2]

Esta tecnologia não permite a interligação em simultâneo de vários dispositivos. A nível de segurança, esta tecnologia tem fragilidades importantes [22, 23].

2.2.2 Wi-Fi

Wi-Fi é uma tecnologia criada pela Wi-Fi Alliance com o objectivo de interligar vários dispositivos sem fios, usando o protocolo IEE 802.11. As infra-estruturas de acesso podem ser públicas ou privadas. Exemplos onde são tipicamente instalados: locais públicos, como aeroportos, centros comerciais e hotéis.

Principais características:

- A frequência de operação é aproximadamente 2.4GHz na Europa e 5GHz na EUA;
- Consumo entre os 10mW e 1W dependendo da frequência utilizada e do país de utilização;
 - O ponto de acesso emite o sinal até uma distância de 300m;
 - Taxa de transferência máxima de 54Mbps.
 - Comodidade, flexibilidade, acessibilidade e mobilidade de acesso à internet.

Principais aplicações:

- Interligação de vários dispositivos;
- Redes de acesso à internet;
- Redes locais internas de escritórios e residências;



Figura 2.3: Comunicação Wi-Fi [3]

Esta tecnologia apesar da grande taxa de transferência de dados possui um consumo energético relativamente elevado [3, 24].

2.2.3 ZigBee

Zigbee é um protocolo de comunicação sem fios baseado no padrão 802.15.4 - 2003, desenvolvido pela ZigBee Alliance em 2004.

O desenvolvimento deste protocolo deveu-se à inexistência de uma norma de redes sem fios aplicada a sensores e a dispositivos de controlo. Contrariamente às tecnologias existentes na altura, (Bluetooh ou WIFI) que permitiam grandes taxas de transferência, a tecnologia ZigBee pretendia a transferência de dados sem fios com a utilização do mínimo consumo de energia e com uma elevada fiabilidade. Outro grande factor para o desenvolvimento foi a tentativa de uma uniformização dos sistemas, a criação de um modelo standard que permitisse a compatibilidade entre sistemas.

Principais Características:

- O baixo consumo (40mA no modo activo e 1μA no modo de standby) e a baixa potência permitem uma longa duração (vários anos) com o uso de pequenas baterias;
 - A sua conectividade em malha de rede permite obter um maior alcance;
 - Suporta um grande número de nós de ligação, possibilitando até 65535 dispositivos;
 - Elevada fiabilidade;
 - Elevada segurança;
 - Alcance até 100 metros;
 - Taxa de transmissão de 250Kb/s;
 - Custo reduzido;
 - Admite diferentes topologias da rede;
- Opera nas frequências de 868 MHz na Europa, 915 MHz em países como EUA e Austrália e também pode operar nos 2.4 GHz.

Principais aplicações:

- Redes de controlo industrial;
- Utilização em computadores e periféricos;
- Controlo residencial e comercial;
- Electrónica de consumo;



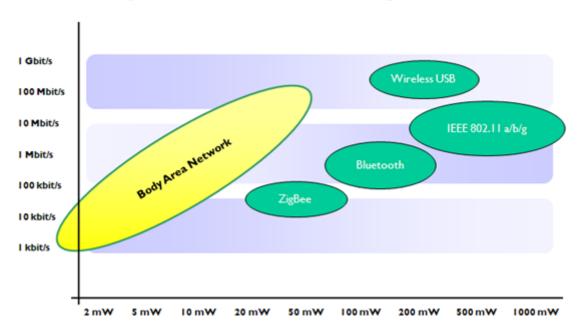


Figura 2.4: Módulo ZigBee [4]

Apesar das vantagens que esta tecnologia oferece, como o baixo consumo e uma taxa de transferência razoável, pretende-se obter ainda um menor consumo, conseguindo simultaneamente uma maior taxa de transmissão de dados para a transmissão de sinais [25, 26].

2.2.4 Resumo

O que se pretende é uma comunicação sem fios que tenha o menor consumo possível e uma taxa de transferência aceitável, 160Kb/s, para a transmissão de um sinal de áudio com boa qualidade. Como se pode observar na figura 2.5 o recurso à comunicação pelo corpo humano é a que permite a melhor performance para os objectivos propostos. Para além disto, tem um alcance bem confinado, o que é essencial em termos de segurança.



Body Area Networks -Target Position

Figura 2.5: Relação consumo/taxa de transferência por tecnologia [5]

2.3 Aplicações de BANs

Devido aos diversos componentes que podem ser conectados e integrados na área corporal as BANs serão capazes de fornecer diversas funções na área da saúde, em situações de emergência, no trabalho, no estilo de vida, no desporto e também em aplicações militares. De seguida descrevem-se alguns exemplos de aplicações.

2.3.1 Monitorização da actividade física

Na prática de desporto o atleta visualiza o desempenho do seu corpo durante toda a sua actividade física, uma vez que tem acesso a vários parâmetros, tais como:

- Frequência cardíaca;
- Temperatura corporal;
- Consumo de oxigénio;
- Pressão arterial;
- Velocidade atingida;
- Distância percorrida.

Os dados adquiridos podem ser guardados num dispositivo, como um mp3, uma vez que é usual a sua utilização na prática de desporto, de uma forma discreta, onde

os dados recolhidos podem ser utilizados para estudo futuro de forma a observar a evolução do atleta, figura 2.6 [5].



Figura 2.6: Atleta com mp3 que visualiza a actividade corporal [5]

2.3.2 Identificação e segurança

As BAN's podem ser utilizadas como meio de identificação pessoal, como por exemplo: autenticação de uma pessoa para o acesso a um sector de uma empresa, acesso a equipamentos ou simplesmente para dar entrada de um funcionário numa empresa, figura 2.7 [6].

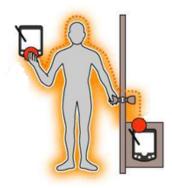


Figura 2.7: Implementação de sistema de identificação [6]

2.3.3 Transmissão de sinais de áudio

As rede BAN permitem a transmissão de dados sem o recurso de fios, por exemplo uma pessoa poderá ouvir confortavelmente música sem ter de usar fios para a comunicação do sinal de áudio do leitor de música até aos auscultadores, figura 2.8 [5]. Estes dispositivos poderão ser:

- Telemóveis;
- Leitores de mp3;

- Leitores de cd's;
- Kits de mãos livres
- Gadjects.



Figura 2.8: Transmissão de sinal de áudio [5]

2.3.4 Rede de sensores do corpo humano

Neste caso os vários sensores que se encontram ao longo do corpo humano têm como principal finalidade a monitorização dos sinais vitais de um paciente, figura 2.9, tais como:

- Pressão sanguínea;
- Temperatura corporal;
- ECG;
- EEG;
- Frequência respiratória;
- Sons tóraxicos.

Com o aumento da esperança média de vida, o número de idosos tende a aumentar. Estes carecem de maiores cuidados de saúde e com a utilização da rede BAN na sua residência, qualquer alteração anormal no seu estado fisiológico poderá logo ser detectado permitindo de imediato a prescrição de tratamento.

Além da monitorização, as BAN's também são relevantes para o diagnóstico precoce e tratamento de doenças crónicas, tais como doenças cardiovasculares, hipertensão e diabetes.

Em última análise, as BAN's permitem um grande desenvolvimento no sistema de saúde personalizado, onde o tratamento pode ser adaptado quer ao nível da monitorização, detecção e diagnóstico [5].

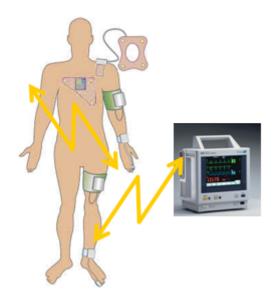


Figura 2.9: Monitorização de uma rede de sensores [5]

2.3.5 Situações de emergência

As BAN'S também podem ser usadas em detecção de casos de emergência. Em caso de acidente, o sistema detecta alteração do estado fisiológico do corpo humano e informa o hospital do sucedido e envia os dados para a ambulância, figura 2.10. Desta forma tornava a assistência ao paciente mais rápida podendo salvar vidas.



Figura 2.10: Diagrama de comunicação entre um paciente e o serviço de urgência [7]

Este sistema aplicado nas cooperações de bombeiros seria uma preciosa ajuda. O comandante poderia deliberar as suas situações de uma melhor forma uma vez que possuía mais informações, dado que poderia verificar o estado actua de uma dado bombeiro e a sua posição, figura 2.11. Além disso todas as informações recolhidas poderiam ser guardadas para uma posterior análise e diagnóstico [8].



Figura 2.11: Bombeiro equipado com sistema BAN [8]

2.3.6 Aplicações Militares

Um soldado que possua um equipamento militar com um sistema BAN incorporado consegue obter informações que podem ser cruciais em campo de batalha, uma vez que este fica interligado a diversos sensores tais como os sensores de monitorização da actividade corporal e a dispositivos como PDA's e GPS.

Em situações de combate a rede BAN pode indicar informações sobre o ambiente que rodeia o soldado, a sua condição fisiológica, permitir distinguir os alvos dos elementos da sua equipa prevenindo o disparo sobre estes.

A base e centro de comando podem estar em constante monitorização das suas tropas podendo entrar em contacto com cada soldado para dar instruções e informar das suas estratégias, bem como o soldado poderia entrar em contacto em pedido de auxílio [8].

2.4 Projectos BAN reportados na literatura

Como se trata de um assunto relativamente recente, não há muita literatura sobre o assunto. De qualquer maneira segue-se seguidamente um resumo dos diversos projectos

desenvolvidos ou em desenvolvimento.

2.4.1 RedTacton

Este projecto tem sido desenvolvido pela NTT (Nippon Telegraph and Telephone Corporation). Neste caso a comunicação é efectuada através do uso de campo eléctrico no qual é possível a comunicação através de qualquer parte do corpo com um taxa de transferência até 10 Mbps.

Para efectuar uma comunicação o transceiver emissor induz um campo eléctrico na superfície do corpo humano onde os dados transmitidos são emitidos através de flutuações do campo eléctrico, criando alterações no estado deste. O transceiver receptor é constituído por um cristal electro-óptico e por uma fonte de luz laser que irá detectar as alterações do campo eléctrico através de um sensor de campo eléctrico combinado, figura 2.12 [6].

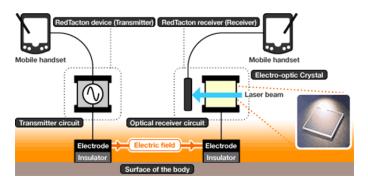


Figura 2.12: Diagrama de blocos do modo de funcionamento [6]

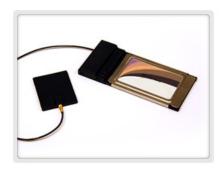


Figura 2.13: Protótipo RedTacton desenvolido [6]

2.4.2 ETRI

O objectivo deste projecto tem como base a transferência de documentos para uma impressora com uma taxa de transferência de 1 Mb/s quando se toca nesta com um dedo.

Neste momento o protótipo desenvolvido só permite ainda uma taxa de transferência de 2.4Kb/s, figura 2.14 [9].



Figura 2.14: Protótipo ETRI [9]

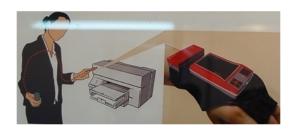


Figura 2.15: Principio de funcionamento do protótipo ETRI [9]

2.4.3 Transceivers existentes

2.4.3.1 Zimmerman

Este foi o primeiro protótipo desenvolvido, criado em 1995. Este tranceiver foi projectado para transmissões com um consumo muito baixo e permitir comunicações half-duplex, contudo, este é fortemente influenciado pelas condições externas e está limitado no máximo a uma taxa de transferência de $2.4 \mathrm{Kb/s}$.

Este transceiver é controlado através de um microcontrolador, nomeadamente um PIC (Microchip) que controla todas as operações. Para efectuar uma transmissão, o microcontrolador envia o sinal digital para um bloco ressoante LC de forma a gerar um sinal sinusoidal.

O sinal recebido detectado atravessa pelo bloco de recepção que é composto por um amplificador de transimpedância, seguido de um amplificador de tensão de ganho variável. De seguida através do controlo de um interruptor analógico o sinal poderá ser invertido. Por fim o sinal atravessa um integrador para que ao entrar na ADC do microcontrolador, este possa detectar o estado lógico do *bit* do sinal recebido (ver figura 2.16).

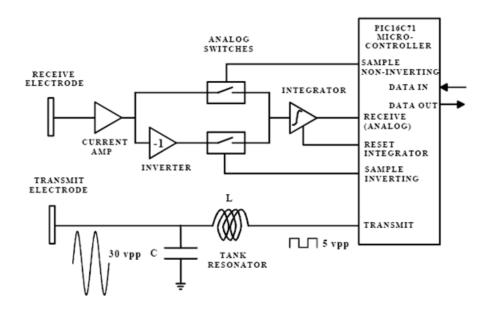


Figura 2.16: Diagrama de blocos Zimmerman [10]

O protótipo desenvolvido tem as dimensões 94 mm x 120 mm x 37 mm e é alimentado por duas baterias de 9V, com um consumo de 400mW, figura 2.17 [10].



Figura 2.17: Protótipo Zimmerman desenvolvido [10]

2.4.3.2 Hachisuka

Os transceivers projectados por Hachisuka em 2003 possuem pequenas dimensões, baixo peso e são alimentados com baterias de 3V. Numa primeira versão os transceivers desenvolvidos efectuam a modelação FM e na versão seguinte efectuam a modelação FSK, figura 2.18 [11].

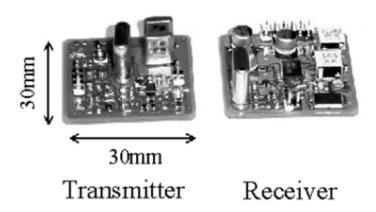


Figura 2.18: Protótipo Hachisuka [11]

2.4.3.3 Shinagawa

Este transceiver possibilita a comunicação half-duplex, com uma taxa de transmissão de $10 \,\mathrm{Mb/s}$ com uma taxa de erro de 0.04% por cada $1070 \ bytes$.

Os sensores até à data permitiam uma taxa de transferência na ordem dos 40Kb/s até uma distância de 30cm. Esta limitação devia-se à resistência de entrada dos sensores não ser suficientemente alta. Para melhorar estes aspectos este transceiver foi projectado com uma grande resistência de entrada e uma grande largura de banda o que é extremamente útil para a detecção precisa de campos eléctricos de pequenas intensidades e rápidos, possibilitando uma taxa de transferência de 10Mb/s até uma distância de 150cm.

Uma nova funcionalidade incorporada foi a possibilidade de transmissão de sinais entre várias pessoas através da roupa ou através de um simples aperto de mão.

As dimensões do transceiver são $15 \times 55 \times 80 \,\mathrm{mm}$, alimentado por uma bateria de $5\mathrm{V}$ e com um consumo de $650 \,\mathrm{mW}$.

2.4.3.4 Song

Este transceiver criado em 2007 foi desenvolvido para baixa potência para sinais de banda larga (WBS). Para efectuar esta comunicação este transceiver utiliza uma interface de acoplamento directo ao corpo humano com uma impedância de 50Ω , um circuito CDR ($clock\ data\ recovery$) que incorpora um VCO e um circuito QSPD ($quadratic\ sampling\ phase\ detector$) que permite um baixo consumo. No entanto a necessidade de baixa impedância de acopolamente (50Ω) obriga a utilização de eléctrodos Ag-AgCl pouco práticos, pelo que não é considerado nesta tese com mais detalhe.

2.4.3.5 Resumo

Através da análise da tabela 2.1 pode-se comparar as várias características e performance dos vários *transceivers* anteriormente abordados.

	Zimmerman 1995	Hachisuka 2003	Shinagawa 2004	Song 2007	
Método de Comunicação	Banda estreita	Banda estreita Conversão EC		Banda larga	
Eléctrodos	Sinal e Referência	Sinal e Referência	Sinal e Referência	Sinal	
Modulação	OOK/DSSS	FM/FSK	Sem modulação	Sem modulação	
Frequência da Portadora	330 KHz	10.7 MHz	0 - 10 MHz	1 - 200 MHz	
Taxa de Transmissão	2.4 Kb/s	9.6 Kb/s	10Mb/s	2Mb/s	
Tensão de Alimentação	9V	3V	5V	1V	
Consumo	400mW	Não referido	650mW	5mw	
Energia por bit	170µJ	Não referido	65nJ	2.5nJ	

Tabela 2.1: Comparação dos vários transceivers desenvolvidos [20]

Dos vários transceivers analisados, não se poderá definir o melhor, uma vez que tem de se ter em conta a aplicação desejada. Neste caso o que se pretende é um compromisso entre a taxa de transferência, consumo e dimensões. Em termos absolutos o melhor é o transceiver desenvolvido por Song em 2007, dado que este possibilita uma taxa de transferência de 2Mb/s bastante superior ao pretendido (160Kb/s), possuir um consumo relativamente baixo na ordem dos 5mW e utilizar unicamente um único eléctrodo. No entanto como não é capacitivo, não se pode comparar directamente com os outros.

2.5 Propagação de sinais através do corpo humano

Nesta secção irão ser abordados alguns estudos [12, 11, 20, 21] efectuados sobre a propagação de sinais através do corpo humano. Estes estudos dão informações sobre os resultados previstos, que posteriomente servem para comparação com os obtidos deste trabalho.

No primeiro estudo [12] foram efectuados vários testes, mais propriamente, a resposta em frequência e no tempo a indivíduos com idades compreendidas entre o 24 e 32

anos, altura variando desde os 1.5 a 1.8m e peso de 45Kg a 80Kg numa sala onde apenas se encontravam cadeiras e mesas. Estas medições foram efectuadas com o auxílio de um *network analyzer*, figura 2.19.

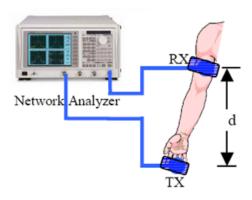


Figura 2.19: Configuração utilizada [12]

As medições foram efectuadas variando os parâmetros que se encontram na tabela 2.2.

Parâmetro	Valor
Gama de frequências [MHz]	1 - 3000
Número de amostras	1201
Potência transmitida [dBm]	15
Distância entre TX e RX [cm]	20 - 55
Número de pessoas testadas	6

Tabela 2.2: Principais parâmetros [12]

Definiram-se 2 modelos de medição. Um em que o endivido se encontra de pé e outro em que se encontra sentado numa cadeira, figura 2.20.

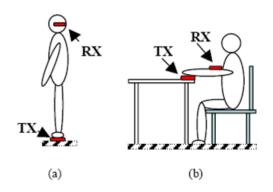


Figura 2.20: Modelos de medida [12]

Comparou-se os testes obtidos pela comunicação pelo corpo humano com os obtidos pela transmissão pelo ar para ambos os modelos (de pé e sentado).

Através da análise dos resultados obtidos nas figuras 2.21 e 2.22 verifica-se que com o aumento da distância entre os eléctrodos a potência do sinal recebido diminui. Verifica-se também que a atenuação aumenta com o aumento da frequência.

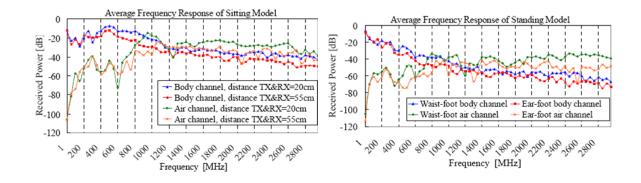


Figura 2.21: Resposta em frequência Figura 2.22: Resposta em frequência quando o individuo se encontra sentado quando o individuo se encontra de pé
[12] [12]

Efectuaram-se também testes com sinal modulado para verificar se se obtía uma melhor resposta. Para efectuar este estudo variaram-se vários parâmetros tais como o tipo de modulação, a frequência da portadora, a distância entre *transceivers*, representado na tabela 2.3.

Parâmetro	Valor		
Tipo de modelação	MSK, BPSK, QPSK		
Frequência da portadora	200, 300, 400, 500		
Taxa de tranmissão [Kb/s]	100 - 2500		
Distância entre transceivers [cm]	20, 155		

Tabela 2.3: Principais parâmetros [12]

Os resultados obtidos encontram-se na figura 2.23, no qual podem retirar-se as seguintes elações:

- A comunicação através do corpo humano permite uma comunicação com menos distorção que comparativamente com a comunicação através do ar, uma vez que o corpo humano não é tão afectado por interferências externas e pelo ruído;
- Com o aumento da distância entre *transceivers* e da taxa de transmissão verificouse um ligeiro aumento de distorção em todos os tipos de modelação;

- A modelação BPSK apresenta uma boa EVM (*Error Vector Magnitude* - métrica que quantifica a diferença entre a resposta ideal e a medida, quanto menor for esta relação melhor foi o resultado obtido) quer quando a distância entre transceivers é pequena (22cm) ou aumenta (155cm) e taxas de transmissão testadas. A modelação MSK apresenta uma boa EVM excepto quando a distância dos transceivers aumentou para 155cm. A modelação QPSK apresentou resultados que informam que não é apropriada para este tipo de comunicação através do corpo humano [12].

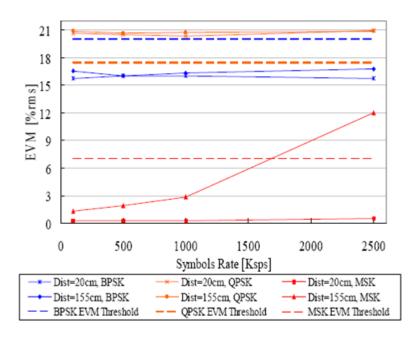


Figura 2.23: Resposta EVM em função de vários factores [12]

Outro estudo [11] efectuado com a transmissão de sinais pelo corpo humano incide na análise em frequência do corpo humano, ou seja, pretende-se determinar qual a frequência que apresenta uma melhor resposta para transmissão de sinais. Para efectuar este estudo procedeu-se à configuração mostrada na figura 2.24.

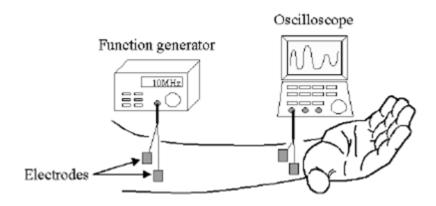


Figura 2.24: Configuração utilizada [11]

Através da análise dos resultados obtidos, figura 2.25, verifica-se que a melhor frequência para a transmissão de sinais é de 10MHz, uma vez que é para esta frequência que a atenuação é menor. Relativamente ao meio de comunicação conclui-se que através do corpo humano a atenuação do sinal é menor relativamente à transmissão pelo ar [11].

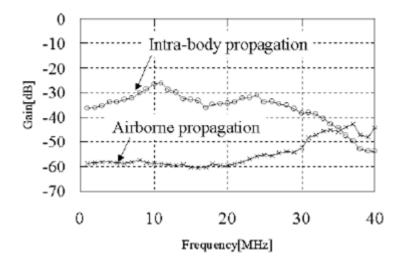


Figura 2.25: Resposta em frequência na comunicação pelo corpo humano e pelo ar [11]

Ao contrário do estudo anterior, Song efectuou um estudo [20] da resposta em frequência no corpo humano usando unicamente um único eléctrodo para a transmissão de sinais, contudo este também apresenta problemas dos eléctrodos Ag-AgCl.

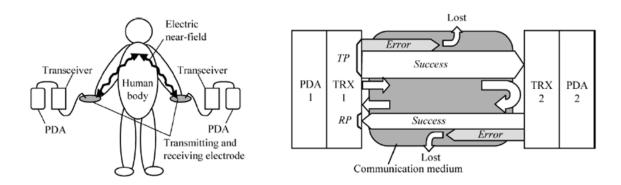
Neste estudo [21], Shinagawa para testar a **fiabilidade** do seu protótipo efectuou vários testes: inicilamente efectuou uma transmissão de um byte ([01010011]) com uma taxa de transferência a 10Mb/s onde os eléctrodos encontravam-se a uma distância de 1m entre eles sendo o sistema alimentado por baterias de forma a reduzir o ruído.

Para efectuar testes à comunicação, uma pessoa está ligada com as mãos aos transceivers que por sua vez estão conectados cada um a um PDA, como mostra a figura 2.26.

O sinal emitido para a superfície do corpo humano é menor que 1V/m de forma a não provocar efeitos no corpo humano.

Confirmou-se a comunicação half-duplex com uma taxa de 10Mb/s em conformidade com IEEE 802.3 através do envio de um comando de confirmação da conexão.

Constatou-se também que é possível a comunicação através da roupa sem a necessidade de estar em contacto com o corpo.



comunicação pacotes [21]

Figura 2.26: Configuração utilizada para a Figura 2.27: Diagrama de transferência de

[21]

Através da análise da tabela 2.4 verificou-se que a taxa de pacotes com erros quando é efectuada a comunicação através do corpo humano é relativamente baixa, na ordem dos 0.03%. Constatou-se que a transmissão de sinais entre duas pessoas era instável e todos os pacotes eram perdidos tendo em conta o pior caso [21].

Tamanho do pacote	UTP	HB1	HB2
560 bytes	0%	0.03%	3%
1070 bytes	0%	0.04%	3%

Tabela 2.4: Taxa de erro de pacotes [21]

UTP - comunicação entre PDAs usando cabo UTP

HB1 - comunicação entre PDAs usando a superfície do corpo humano de 1 pessoa

HB2 - comunicação entre PDAs usando a superfície do corpo humano de 2 pessoas

Para efectuar um estudo semelhante aos referidos anteriormente, foi necessário projectar um setup. Neste setup existe um componente electrónico essencial que torna 2.6. DDS 25

possível efectuar o estudo, o *DDS*. De seguida segue-se uma descrição detalhada deste componente.

$2.6 \quad DDS$

DDS (Direct Digital Synthesis) é um método utilizado para gerar sinais, normalmante sinusoidais, com uma dada frequência e fase a partir de um relógio de referência.

O funcionamento interno digital possibilita uma rápida comutação entre frequências, permite o ajuste de fase na ordem do décimo de grau e uma grande gama de frequências de operação, desde frequências na ordem dos micro hertz até os $500 \mathrm{Mhz}$. O seu grande uso deve-se também ao seu reduzido consumo e tamanho. Contudo a frequência máxima teórica que um DDS pode gerar é metade da sua frequência interna; este facto deve-se ao critério de Nyquist e resulta em que comparativamente aos circuitos baseados em PLL (phase lock loop) não se atingam frequências tão elevadas. De referir que uma melhor performance é obtida para frequências inferiores a 1/3 da frequência de relógio onde o spurious é menor, ou seja, onde a distorção é menor.

Os DDS são facilmente programados através de uma comunicação série síncrona como a comunicação SPI e SSR, uma vez que a maioria dos microcontroladores e microprocessadores possuem esta interface [27, 15].

2.6.1 Arquitectura

De uma forma simplista um DDS é composto por 3 grandes blocos, um acumulador de fase, um conversor de fase para amplitude e um conversor digital analógico (DAC), figura 2.28.

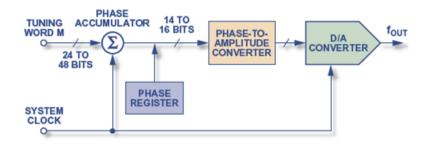


Figura 2.28: Diagrama de blocos simplificado de um DDS [13]

Numa PROM está armazenada um vector/tabela que contém as amplitudes de um período de uma sinusóide com a resolução máxima de funcionamento.

O método para gerar uma onda com uma dada frequência baseia-se em "saltos" sobre os índices da tabela.

A frequência de uma onda é obtida da seguinte forma:

$$f_0 = \frac{FTW \times f_{CLK}}{2^n}$$

Onde:

FCLK corresponde à frequência interna de operação do DDS;

n, o número de *bits* do acumulador de fase e FTW ou M, o número índices que tem de se saltar na tabela (tunning word);

 f_0 , frequência de saída do DDS.

O acumulador de fase funciona como um incrementador, onde em cada ciclo de relógio incrementa com o valor de M o índice da tabela calculado no ciclo anterior, determinando desta forma o próximo índice da tabela que contém o próximo valor de amplitude do sinal de saída.

Na figura 2.29 está representado de uma forma ilustrada o conceito de funcionamento do acumulador de fase. Sobre uma roda estão marcados vários pontos em que cada ponto corresponde a um ponto de um ciclo da forma de onda gerada. Em cada ciclo de relógio o índice anterior irá ser incrementado por um valor constante, M, que irá determinar o novo ponto na roda, ou seja, o próximo valor de amplitude do sinal. Assim quando se percorrer uma volta completa da roda, acabou de ser gerado um período de onda. O número de pontos dispostos ao longo da roda irá depender do número de bits de resolução, n, do DDS.

Quanto maior for o valor de M, mais rapidamente se percorre a tabela, ou seja, obtêm-se uma sinusóide com uma maior frequência.

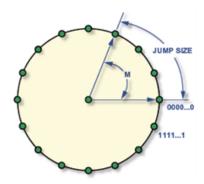


Figura 2.29: Roda ilustrativa [13]

Os índices de saída do acumulador de fase correspondem a amostras de um sinal rampa. Estes são convertidos num sinal sinusoidal amostrado através do uso da tabela

2.6. DDS 27

que contêm os valores de amplitude do período de um sinal sinusoidal de referência, o que por sua vez é convertido num sinal analógico através de uma DAC, figura 2.30.

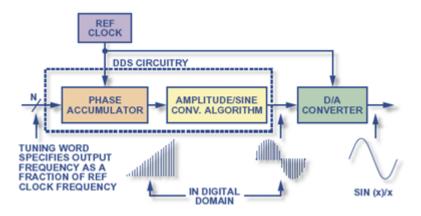


Figura 2.30: Vários passos para criação de um sinal [13]

O DDS tem uma funcionalidade que permite utilizar vários DDS recorrendo unicamente a um único relógio de referência. Com esta capacidade de poder utilizar vários DDS sincronizados, tornando possível realizar modulação I-Q (fase e quadratura), figura 2.31. Alguns DDS já têm incorporado funcionalidades que permitem outros tipos de modulação como FSK e PSK [27, 15, 14, 13].

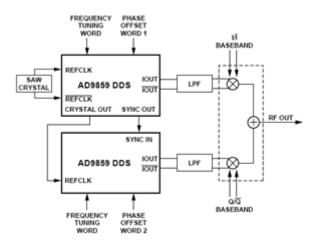


Figura 2.31: DDS sincronizados para modulação I-Q [14]

2.6.2 Jitter

O Jiiter é um factor importante e indesejável nos circuitos electrónicos. Este factor corresponde à variação temporal de uma dada característica de um sinal aleatório.

Num oscilador as transições ascendentes e descendentes ocorrem em momentos precisos do tempo que nunca variam, contudo, na realidade esta situação é muito difícil de suceder, uma vez que, os osciladores são constituídos por componentes reais, que entre outros potenciais problemas possibilitam a introdução de ruído. Este ruído pode ser térmico, proveniente da fonte de alimentação devido à influência de campos eléctricos e magnéticos e também resultado da conversão do sinal através de uma DAC.

De forma a reduzir este factor deverá utilizar-se um oscilador de referência com baixo *jitter* e efectuar médias uma vez que este tem um caracter aleatório [15].

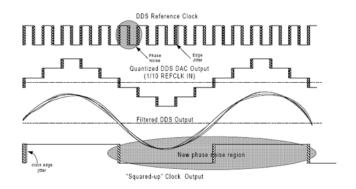


Figura 2.32: Redução da influência do jitter no sinal de saída do DDS [15]

Em alguns DDS, tal como o DDS AD9859 da Analogue Devices [14] utilizado neste trabalho são programados através da interface SPI.

2.6.3 SPI

O protocolo SPI é um meio de comunicação de dados síncrono criado pela Motorola que opera no modo $full\ duplex$. Neste protocolo a comunicação é feita entre master/slave onde quem inicia sempre a comunicação é o master.

Na comunicação SPI existe sempre uma troca de dados entre dispositivos, sempre que são enviados dados, novos dados são recebidos.

A frequência de transmissão pode variar desde 1MHz a 70MHz.

O barramento SPI é composto por 4 sinais:

- SCLK Serial Clock;
- MOSI Master Output, Slave Input;
- MISO Master Input, Slave Output;
- SS Slave Select.

2.6. DDS 29

Frequentemente também são utilizados outros nomes para estes sinais:

- SCK Serial Clock;
- SDI Serial Data In;
- SDO Serial Data Out;
- nCS, CS, nSS, STE Chip Select, Slave Transmit Enable

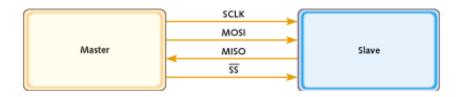


Figura 2.33: Comunicação SPI entre Master/Slave [16]

A comunicação SPI pode ser efectuada com um master e um ou mais slaves. No caso em que se usam vários slaves o barramento é partilhado (SCLK, MISO, MOSI), ou seja, permite uma redução do número de ligações. Normalmente os dispositivos têm incorporado saídas tri-state, pelo que quando o dispositivo não está activado as suas saídas ficam em alta impedância. Caso o dispositivo não tenha incorporado esta funcionalidade não poderá partilhar o barramento.

O modo SPI define a polaridade e o nível da transição do sinal de *clock*. A polaridade do sinal de *clock* refere se o sinal de *clock* no seu estado *idle* se encontra no estado lógico *high* ou no estado lógico *low*. A transição do sinal de *clock* identifica se a aquisição dos dados é adquirida quando o sinal de *clock* efectua uma transição ascendente ou descendente.

Geralmente nos microcontroladores PIC18 da Microchip [16] para configurar o modo de funcionamento da interface SPI são utilizados dois registos, o CKP e o CKE. O registo CKP configura a polaridade do sinal de *clock* enquanto o registo CKE configura em que transição os dados são adquiridos.

Na tabela que se segue encontram-se os vários modos existentes para o caso concreto do PIC18LF458 da Microchip [16].

Modos SPI	CKP	CKE
0,0	0	1
0,1	0	0
1,0	1	1
1,1	1	0

Tabela 2.5: Modos SPI existentes [16]

Nota: os modos mais utilizados nas comunicações SPI são o modo (0,0) e o modo (1,1).

Para iniciar uma comunicação o master selecciona com qual dos slaves pretende efectuar uma comunicação, colocando o sinal (\overline{SS}) referente a esse slave no estado low.

Com o slave activo, o master envia os dados previamente armazenados num buffer bit a bit por cada ciclo de relógio para a sua linha SDO e ao mesmo tempo que está a enviar dados está também a receber pela sua linha SDI.

Quando a transmissão de dados finalizar o master coloca a linha (\overline{SS}) referente ao slave que acabou de comunicar no estado high [28, 16].

2.7 Network analyzer

Os network analyzer são instrumentos utilizados para analisar as características de dispositivos, especialmente associadas com a transmissão e reflexão de sinais eléctricos, tais como a impedância, VSWR, ganho, perdas por retorno, atraso de grupo, podendo estes operar desde uma faixa de frequência inferior a 1Hz até os 110GHZ. Possuem uma gama dinâmica tipicamente de 100dB podendo ir até os 130 dB.

Com este instrumento é possível analisar o comportamento linear e não linear de muitos dispositivos quer estes sejam passivos ou activos, tipicamente são disponibilizados entre 2 e 4 portos, mas há aparelhos com 8 portos.

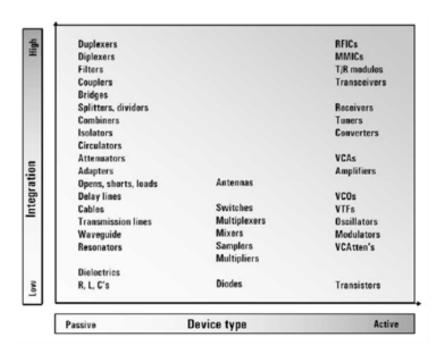


Figura 2.34: Lista de dispositivos possíveis de analisar com o network analyzer [17]

Os network analyzer podem ser subdivididos em duas categorias. Os vector network analyzers que permitem medir a amplitude e fase, enquanto os scalar network analyzers permitem apenas medir a amplitude.

Um network analyzer é constituído por 4 grandes blocos:

- A fonte de sinal permite excitar o circuito de teste de forma a poder medir a resposta deste.
- O bloco de separação de sinal tem 2 principais objectivos, separar as ondas incidentes e reflectidas e medir a onda incidente recorrendo a *spliters* ou a *directional couplers* para que esta possa ser utilizada como referência.
- O bloco detector/receptor detecta o sinal RF de duas maneiras possíveis, ou através de um díodo/filtro no qual converte o sinal RF para um sinal DC, através da remoção da componente AC, ou através do uso de um oscilador local e de um mixer que converte o sinal RF para um sinal com uma frequência intermédia, sendo este filtrado de seguida por um filtro passa banda para eliminar a frequência imagem e também para que o receptor não necessite de uma grande largura de banda permitindo uma maior selectividade. Adicionalmente os network analyzer mais recentes têm incorporado uma ADC e um DSP de forma a poder extrair a fase e a amplitude do sinal resultante.

Por fim no **bloco de processamento/amostragem** os dados adquiridos são tratados e processados e de seguida os resultados obtidos são visualizados através de um display de uma forma user-friendly [17, 29].

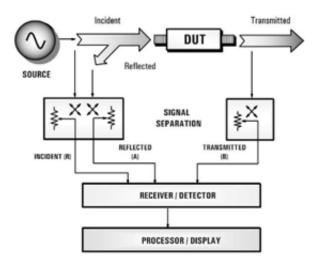


Figura 2.35: Diagrama de blocos de um vector network analyzer [17]

Para caracterizar completamente um dispositivo é necessário medir a fase e a amplitude em diversas condições determinando um conjunto de parâmetros que descrevem o comportamento eléctrico do dispositivo em quaisquer condições, criando desta forma um modelo do dispositivo.

2.7.1 Parâmetros S

De vários tipos de parâmetros passíveis de serem determinados, em alta frequência são tipicamente usados os parâmetros S, uma vez que ao contrário de outros que são calculados através da medição da corrente e de tensão nos portos do dispositivo com a utilização de curto-circuitos ou de circuito-aberto de difícil realização em alta frequência. A medição recorrendo aos parâmetros S é relativamente mais fácil, uma vez que basta medir as ondas de tensões nos portos, recorrendo para tal uma carga resistiva, normalmente de 50Ω , não sendo necessário a utilização de curto-circuitos nem de circuitos abertos que podem provocar oscilações [17, 30, 31].

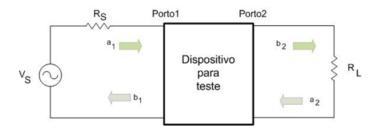


Figura 2.36: Diagrama de reflexões

As 2 equações que descrevem o sistema são:

$$b_1 = S_{11}a_1 + S_{12}a_2$$

$$b_2 = S_{21}a_1 + S_{22}a_2$$

$$S_{11} = \frac{b_1}{a_1}, \ para \ a_2 = 0, \ ou \ seja, \ Z_L = Z_0 \quad S_{12} = \frac{b_1}{a_2}, \ para \ a_1 = 0, \ ou \ seja, \ Z_S = Z_0 \\ S_{21} = \frac{b_2}{a_1}, \ para \ a_2 = 0, \ ou \ seja, \ Z_L = Z_0 \quad S_{22} = \frac{b_2}{a_2}, \ para \ a_1 = 0, \ ou \ seja, \ Z_S = Z_0$$

onde,

 Z_0 , Impedância caracteristica da linha

 Z_L , Impedância de carga

A matriz de parâmetros S é dada por :
$$s = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$$
, onde,

 S_{11} , Coeficiente de reflexão na entrada;

 S_{12} , Coeficiente de transmissão para trás;

 S_{21} , Coeficiente de transmissão para a frente;

 S_{22} , Coeficiente de reflexão na carga.

Quando um dos portos do circuito se encontra terminado com a impedância característica a onda incidente irá ser toda absorvida pela carga, não havendo reflexão, assim sendo, sempre que Z_L ou Z_S sejam iguais a Z_0 não existirá onda reflectida nesse porto.

 $\Gamma_S = S_{11} = \frac{Z_S - Z_0}{Z_S + Z_0}$, coeficiente de reflexão na entrada.

 $\Gamma_L = S_{22} = \frac{Z_L - Z_0}{Z_L + Z_0}$, coeficiente de reflexão na carga.

Coeficiente de transmição = Coeficiente de reflexão + 1

De uma forma simplista, os *network analyzer* o que fazem na realidade é comparar o sinal produzido com o sinal reflectido à entrada do dispositivo para teste.

2.7.2 Varrimento na frequência

Recorrentemente é normal ocorrerem problemas na medição em dispositivos com um atraso significativo. Se se efectuarem varrimentos rápidos na frequência a amplitude da resposta é atenuada e aparenta estar distorcida enquanto para varrimentos mais lentos na frequência a resposta obtida já não apresenta esses efeitos.

Quando se mede em equipamentos com um atraso eléctrico significativo ao efectuarse um varrimento na frequência surgirá um tempo de atraso significativo provocando uma variação (ΔF) entre as frequências do sinal produzido e recebido. Assim, quando se compara o sinal inicialmente produzido com o sinal recebido estes terão frequências diferentes, provocando um erro na medição da amplitude e fase [18], figura 2.37.

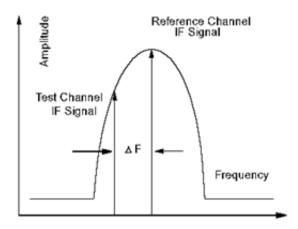


Figura 2.37: Variação de frequência entre o sinal emitido e recebido [18]

Para poder minimizar este problema dever-se-á utilizar uma taxa de varrimento na frequência dependendo do dispositivo a ser utilizado.

A partir deste momento a tese está estrutura em 3 capítulos, onde é (i) abordado todo o *hardware*, (ii) *software* desenvolvido e (iii) onde se apresentam os resultados obtidos.

Capítulo 3

Hardware

Neste capítulo irá ser feita uma descrição pormenorizada de todo o *hardware* desenvolvido que compõe o sistema, nomeadamente os circuitos motherboard, pré amplificador de entrada e fonte de alimentação, como se esquematiza na figura 3.1.

3.1 Motherboard

Esta placa foi desenvolvida com o objectivo de produzir um sinal sinusoidal com frequência programável até os 166MHz com uma grande resolução de frequência, com uma amplitude variando desde os -64dBV até aos 0.5dBV com um passo de 0.5dBV, enviar esse mesmo sinal para o dispositivo em teste e comparar o sinal gerado com o recebido, tudo controlado através de um microcontrolador.

Através da análise do diagrama de blocos da figura 3.1 pode-se observar o princípio de funcionamento da placa motherboard.

Inicialmente o microcontrolador programa o circuito integrado *DDS* para que este gere um sinal sinusoidal com a frequência pretendida. O sinal gerado atravessa um filtro passa-baixo de forma a eliminar os harmónicos e algum ruído existente. De seguida o sinal é subdividido em duas partes. Numa, o sinal é amplificado/atenuado e enviado para a placa pré amplificador de entrada. Na outra, o sinal é atenuado para que este que este ao entrar no detector de sinal como referência, tenha a amplitude que permita ao detector abranger a maior gama dinâmica possível.

Dois dos detectores irão ter como sinal de referência este sinal enquanto os outros dois irão ter como sinal de referência um sinal proveniente do *DUT*. Este facto deve-se à forma como são determinados os parâmetros S.

Os detectores irão comparar o sinal de referência com um sinal proveniente do DUT, gerando dois sinais DC com amplitude em função da diferença de fase e de amplitude

entre eles.

Com o auxílio de um *switch* é seleccionado qual dos sinais proveniente dos detectores se pretende estudar. Dependendo da frequência que se pretende analisar o outro *switch* irá comutar fazendo com que o sinal seleccionado atravesse por um filtro passa baixo com uma dada frequência de corte para que a componente AC que corrompe o sinal seja removida, não provocando erros na leitura quando este for convertido na ADC do microcontrolador. Está incorporado um lcd neste sistema onde os resultados obtidos poderão ser visualizados.

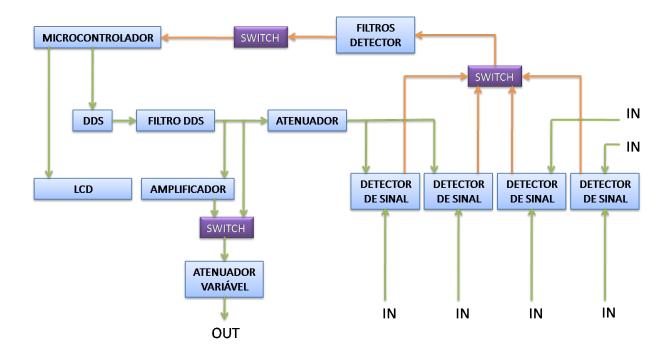


Figura 3.1: Diagrama de blocos da motherboard

3.1.1 Microcontrolador

Para efectuar todo o controlo do vário hardware utilizado recorreu-se a um microcontrolador, nomeadamente o PIC18LF458 da Microchip [16]. Optou-se por este microcontrolador devido ao número de pinos de I/O e ao facto de este poder ser alimentado com uma tensão de 3.3V, sendo este um requisito fundamental dado que a tensão recomendada pelo DDS utilizado para os pinos de I/O com o microcontrolador ser de 3.3V. Na figura 3.2 encontra-se a configuração utilizada para o microcontrolador. Nesta mesma figura pode ser visualizada a interface série entre o microcontrolador e um PC.

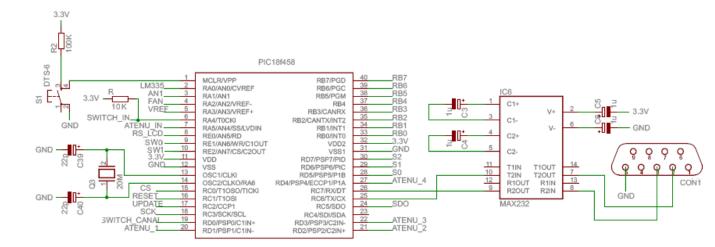


Figura 3.2: Configuração utilizada para o PIC

3.1.2 DDS

O DDS utilizado foi o AD9859 da Analogue Devices [14], este é programado pelo microcontrolador através da interface SPI com a frequência desejada e com os restantes parâmetros de configuração. A configuração utilizada pode ser visualizada na figura 3.3.

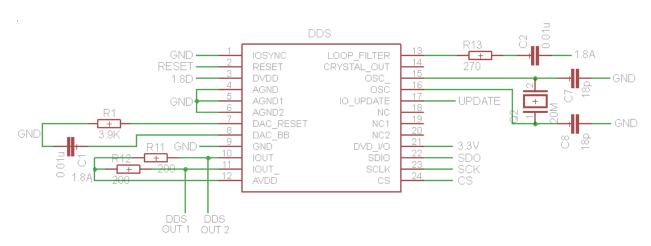


Figura 3.3: Configuração DDS

$3.1.3 \quad DDS \text{ AD9859}$

Optou-se por utilizar neste trabalho um *DDS* AD9859 da Analogue Devices [14] uma vez que este permite uma frequência de relógio interno até 400MHz, ou seja, permite gerar sinais com frequência até 200MHz, teoricamente portanto acima do pretendido. Possui uma resolução de 32 *bits* para a tabela de amplitudes, permitindo

desta forma gerar sinais com uma resolução inferior a $0.1 \mathrm{Hz}$ e possui uma resolução de 14~bits para o offset de fase, ou seja, permite um ajuste de fase menor que um décimo de grau.

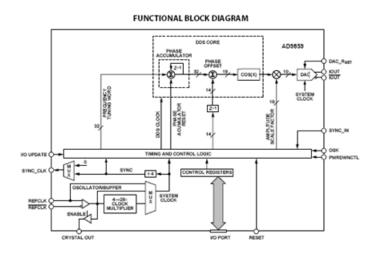


Figura 3.4: Diagrama Funcional do *DDS* AD9859 [14]

Registos internos de AD9859

No AD9859 existem 6 registos que podem ser configurados:

- CFR1, 4 bytes. Configura vários modos, características e funções do dispositivo;
- CFR2, 3 bytes. Permite seleccionar várias funções do dispositivo relacionadas com a parte analógica;
- ASF, 2 bytes. Possibilita efectuar o modo como pode ser feita a modulação em amplitude;
 - ARR, 1 byte. Configuração da modulação em amplitude;
- FTW0, 4 bytes. Define o salto entre amostras da tabela de referência para gerar a frequência pretendida;
 - POW0, 2 bytes. Permite a introdução de um offset de fase.

Protocolo de comunicação de AD9859

A comunicação com o *DDS* é composta em 2 fases. A fase 1 é composta por um *byte* corresponde ao ciclo de instrução no qual informa se pretende efectuar uma escrita ou uma leitura e a que registo se pretende aceder. O primeiro *byte* enviado é composto da seguinte forma:

MSB	D6	D5	D4	D3	D2	D1	LSB
R/\overline{W}	X	X	A4	A3	A2	A1	A0

Tabela 3.1: Composição do primeiro byte de configuração do DDS AD9859 [14]

39

 R/\overline{W} – se pretende efectuar uma leitura ou escrita ;

X - Don't Care;

A4...A0 – endereço série do registo que se pretende aceder.

A fase 2 corresponde ao ciclo de comunicação, no qual os dados são transferidos. Esta fase é composta por um número de *bytes* variável dependendo do registo que se pretende aceder.

O DDS AD9859 tem incorporado uma interface SPI que possibilita a transferência de dados para que este possa ser programado [14].

Como o *DDS* proporciona duas saídas complementares, recorreu-se a um opamp, THS 4631, com a configuração diferença, com o objectivo de remover algum ruído contido no sinal e de colocar o sinal à saída deste com a amplitude de -7.34dBV, figura 3.5.

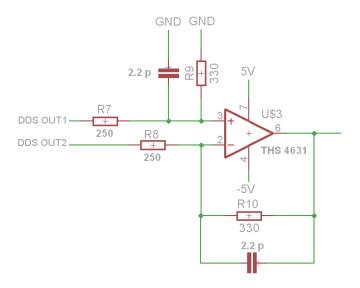


Figura 3.5: Opamp na configuração diferença

O sinal à saída do opamp é dado por:

$$V_0 = \frac{R_9}{R_7} V_{DDS1} - \frac{R_{10}}{R_8} V_{DDS2}$$

Para $R_{10} = R_9 e R_8 = R_7$

$$V_0 = \frac{R_9}{R_7} (V_{DDS1} - V_{DDS2})$$

Como $V_{DDS1} = V_i.cos(wt)$ e $V_{DDS2} = V_i.cos(wt + \pi)$

$$V_0 = \frac{R_9}{R_7} Vi.(cos(wt) - cos(wt + \pi))$$

Através das propriedades trignométricas: $cos(wt + \pi) = -cos(wt)$

$$V_0 = \frac{R_9}{R_7} Vi.(cos(wt) - (-cos(wt)))$$
$$V_0 = 2.Vi.\frac{R_9}{R_7}.cos(wt)$$

Da demonstração efectuada anteriormente poderá constatar-se que ao efectuar a diferença entre os dois sinais, o sinal resultante irá ser um sinal com a mesma frequência de ambos os sinais provenientes do *DDS*, mas com o dobro da amplitude.

O sinal gerado pelo DDS tem uma amplitude de -16.16dBV e como se pretende que este tenha uma amplitude de -7.34dBV, R_9 terá de ser igual a $1.38 \times R_7$.

3.1.4 Filtro DDS

De seguida o sinal proveniente do opamp é filtrado por um filtro passa baixo com frequência de corte de 166MHz com a finalidade de eliminar os harmónicos do sinal e remover algum ruído existente, figura 3.6.

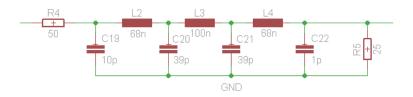


Figura 3.6: Filtro projectado

Este filtro foi projectado recorrendo ao programa AADE Filter Design. Com o uso deste mesmo programa efectuou-se um estudo em frequência do filtro projectado e efectuou-se também uma análise Monte Carlo, como mostram as figuras 3.7 e 3.8.

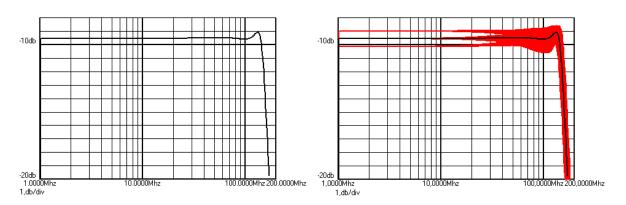


Figura 3.7: Resposta do filtro

Figura 3.8: Análise de Monte Carlo do filtro

41

3.1.5 Atenuador

O sinal proveniente do filtro é subdividido em duas partes e numa delas o sinal atravessa por dois atenuadores independentes com o intuito de provocar uma atenuação de 19.45dBV, figura 3.9, para que o sinal resultante ao entrar no detector de sinal possua uma amplitude de -43dBV, uma vez que é para este valor de tensão que o detector funciona com a maior gama dinâmica possível que é de 60dBV.

Este atenuador foi projectado contendo mais componentes do que os necessários (resistências de 100 Ω) para obter o mesmo efeito. Decidiu-se por esta opção de forma a não sobrecarregar a linha de transmissão. A escolha do valor das resistências de 100 Ω teve em conta o valor da impedância característica da linha.

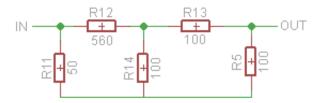


Figura 3.9: Atenuador de 19.45dBV

3.1.6 Detector de sinal

O detector de sinal escolhido foi o AD8302 da Analogue Devices [19]. Este dispositivo permite medir a relação de amplitudes e a diferença de fase entre 2 sinais RF independentes numa gama de frequências desde DC até 2.7GHz, a gama dinâmica de -30dB a +30dB (num sistema de 50Ω), o que significa que permite medir a amplificação e a atenuação de um sinal relativamente ao sinal de referência. Este detector de sinal tem uma boa precisão, com um erro menor que 0.5 dB na medição de amplitude e inferior a 1 grau na medição de fase.

3.1.7 Funcionamento

O AD8302 é constituído pelos seguintes blocos: 2 amplificadores logarítmicos, detector de fase, amplificador de saída, célula de polarização e um *buffer* de tensão de referência, figura 3.10.

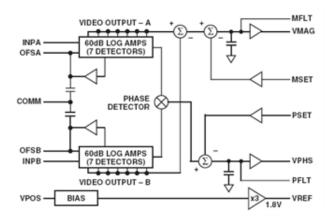


Figura 3.10: Diagrama de blocos AD8302 [19]

O sinal de entrada e o sinal de referência atravessam um detector logarítmico (cada um tem o seu) com uma gama dinâmica de 60dB. Com o uso destes detectores/conversores, para calcular o rácio entre os sinais, efectuam-se diferenças devido à propriedade dos logaritmos.

$$log(a/b) = log(a) - log(b)$$

Com a conversão dos sinais, para se obter na saída a razão dos sinais, basta efectuar a sua subtracção. Caso não se tivesse efectuado esta conversão, para obter o mesmo resulto iria ser necessário efectuar operações que exigem circuitos mais complexos.

O amplificador logaritmico consegue detectar sinais com amplitude desde -60dBm até 0dBm. De forma a cobrir toda a gama dinâmica, o sinal de referência deverá ter um nível de -30dBm, assim sendo, o sinal deste poderá variar a sua amplitude desde os -30dBm a +30dBm.

Relativamente à diferença de fase este detector só detecta dois quadrantes, ou seja, de 0 a π , não informando se o sinal de teste se encontra em avanço ou em atraso relativamente ao sinal de referência.

Perante os sinais de entrada o detector de sinal irá produzir um sinal DC consoante a diferença de amplitude e outro para a diferença de fase. Para a diferença de amplitude irá produzir um sinal com a relação de $30 \, \mathrm{mV/dB}$ e para a diferença de fase de $10 \, \mathrm{mV/grau}$.

A diferença de fase varia entre 0 e 180 graus no qual proporciona na saída a relação de $10\,\mathrm{mV/grau}$, figura 3.11.

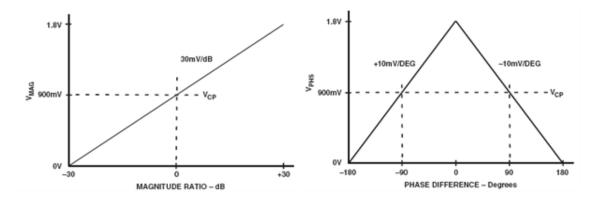


Figura 3.11: Resposta com a variação de amplitude e diferença de fase [19]

Através dos pinos MSET e PSET podem ser seleccionados 3 modos de funcionamento:

- Medição;
- Comparador;
- Controlador.

<u>Medição</u>

No modo de medição o dispositivo limita-se apenas a medir o ganho/atenuação e a diferença de fase entre os dois sinais.

Comparador

No modo de comparador além de efectuar o mesmo que no modo de medição, neste modo o dispositivo delimita o valor da sua saída. Assim à saída do dispositivo obterse-á a seguinte resposta:

 $\begin{cases} 1.8V, se o valor de saída previsto for superior ao valor de referência \\ 0V, se o valor de saída previsto for inferior ao valor de referência \end{cases}$

Controlador

No modo de controlador o dispositivo realiza as mesma funções que os modos anteriores, contudo neste modo o dispositivo funciona como um controlador de malha fechada que estabiliza o ganho/atenuação e a fase do dispositivo em teste, figura 3.12.

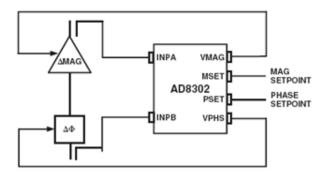


Figura 3.12: Dispositivo operando no modo de controlador [19]

3.1.8 Aplicações

A aplicação mais utilizada é a monitorização do ganho/atenuação e diferença de fase de um dado bloco. Contudo uma aplicação também bastante utilizada é a medição do ganho/atenuação e diferença de fase de ondas incidentes e reflectidas numa dada carga, ou seja, o coeficiente de reflecção [19, 32, 33].

O detector de sinal, AD8302, irá comparar o sinal proveniente do atenuador com o sinal recebido do dispositivo em teste produzindo um sinal DC com amplitude consoante a diferença de fase e outro consoante o ratio entre as amplitudes. Estes estão ligados à entrada da ADC do PIC (10 bits). A configuração utilizada está representada na figura 3.13.

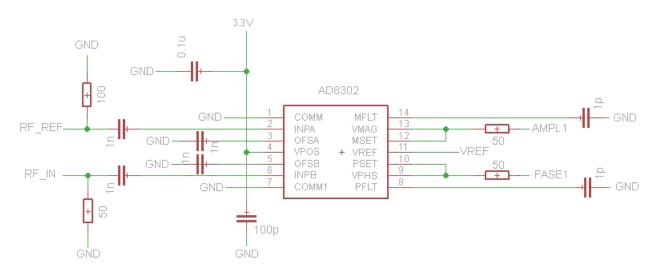


Figura 3.13: Configuração do detector de sinal

Para o estudo do dispositivo em teste irá-se comparar o sinal gerado com o sinal recebido do dispositivo. Este estudo incidirá sobre a análise da diferença de fase e de amplitude entre os dois sinais, determinando um conjunto de parâmetros que descrevem

o comportamento eléctrico do dispositivo em quaisquer condições, mais propriamente os 4 parâmetros S. Para os determinar é necessário analisar 3 sinais provenientes do dispositivo, mais concretamente:

- o sinal reflectido à entrada do dispositivo;
- o sinal incidente na carga;
- o sinal reflectido na carga.

Assim é necessário utilizar 4 detectores de sinal para criar um modelo do dispositivo (o quarto corresponde ao sinal à entrada do DUT).

3.1.8.1 Filtro Detector

A experiência mostrou que o sinal DC proveniente de cada uma das saídas do detector de sinal vem corrompido com uma componente AC com origem do sinal RF de entrada. A existência desta provoca erros de leitura na ADC do microcontrolador e para colmatar este problema projectaram-se vários filtros passa baixo com frequência de corte de 100Hz, 3.3KHz, 100KHz e 3.3MHz, removendo desta forma a componente AC existente no sinal.

Projectaram-se vários filtros (em vez de um único) dado que a componente AC tem origem do sinal RF de entrada e como para um filtro com uma dada frequência de corte, para frequências relativamente superiores, as indutâncias tendem a comportar-se como curto-circuitos, não efectuando a sua função correctamente. Assim sendo, projectaram-se estes filtros com estas frequências de corte para que após uma década o sinal esteja atenuado de 60dB. Na figura 3.14 poderá verificar-se por que filtros atravessa o sinal, em função da frequência do sinal RF de entrada.

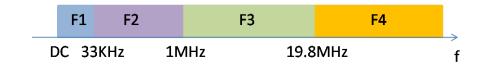


Figura 3.14: Filtros que o sinal atravessa em função da frequência do sinal RF

Nas figuras 3.15, 3.16, 3.17 e 3.18 encontram-se os circuitos dos respectivos filtros implementados. Esta análise além de complementar a tolerância dos diversos componentes, inclui também a resistência induzida da indutâncias.

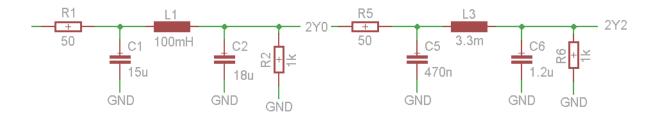


Figura 3.15: Configuração do filtro de Figura 3.16: Configuração do filtro de $100 \, \mathrm{Hz}$ 3.3 KHz

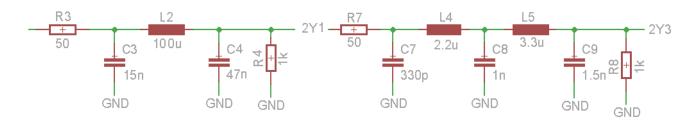


Figura 3.17: Configuração do filtro de Figura 3.18: configuração do filtro de $100 \, \mathrm{KHz}$ 3.3 MHz

Para verificar o comportamento dos vários filtros efectuou-se a cada um deles uma análise de monte carlo, que se encontram nas figuras 3.19, 3.20, 3.21 e 3.22

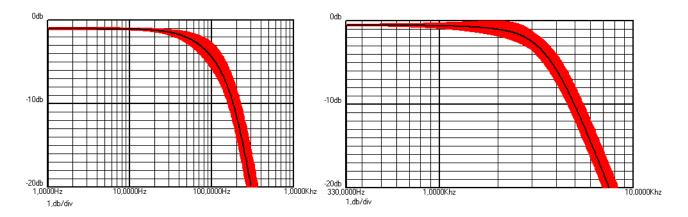


Figura 3.19: Análise de Monte Carlo do filtro de 100Hz

Figura 3.20: Análise de Monte Carlo do filtro de 3.3KHz

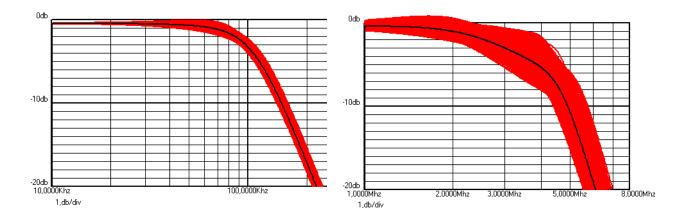


Figura 3.21: Análise de Monte Carlo do Figura 3.22: Análise de Monte Carlo do filtro de 100KHz filtro de 3.3MHz

3.1.9 Selecção do parâmetro a analisar

Utilizou-se um *multiplex* 74HC4051 de 8 entradas para se poder escolher qual o sinal a ser convertida pela adc, amplitude ou fase e a de que parâmetro S correspondente, sendo este controlado digitalmente pelo PIC através de 3 pinos de controlo, como mostra a figura 3.23. Procedeu-se desta forma para maximizar o número de pinos do microcontrolador, em virtude de estes terem sido todos utilizados.

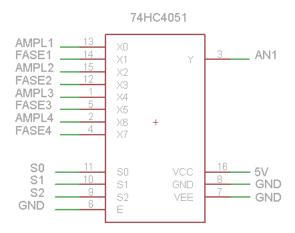


Figura 3.23: Configuração utilizada para o Multiplex

O sinal proveniente do detector de sinal atravessamu pelo multiplex e percorre o filtro corresponde ao canal do multiplex seleccionado. Este sinal posteriormente atravessa pelo desmultiplex e por fim chega à adc do PIC. Para efectuar o mux e o desmux utilizou-se o 74HC4052, onde o controlo é efectuado digitalmente através de 2 pinos de controlo do PIC, como mostra a figura 3.24.

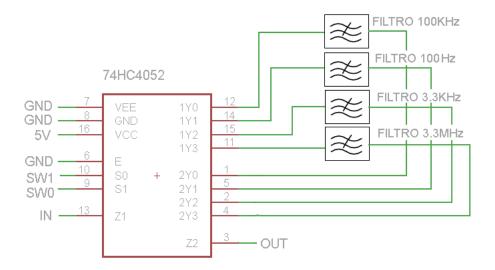


Figura 3.24: Configuração utilizada para o 74HC4052

3.1.10 Circuito Fonte de sinal

Como vimos em 2.7 temos que ter um circuito que excite o dispositivo em teste com um sinal determinado. Neste trabalho este sinal pode ter desde -64dBV até 0.5dBV. Isto é correspondido fazendo passar o sinal à saída do filtro *DDS* (ver figura 3.1) por um amplificador, OPA847 [34], que se encontra na configuração não inversora. Optouse por este amplificador uma vez que este possui um grande ganho x largura de banda, 3.9GHz, mas essencialmente devido a este ser um opamp com um extremo baixo ruído.

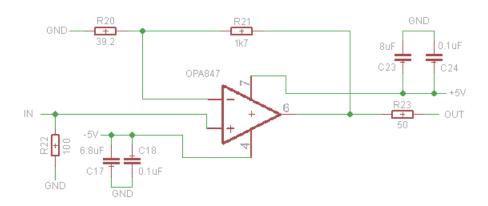


Figura 3.25: Configuração do amplificador

Pretende-se que o opamp proporcione um ganho de 43.48, para que à saída deste se encontre um sinal com amplitude de $0.5 \mathrm{dBV}$

$$A_v = 1 + \frac{R_{21}}{R_{20}}$$

Para $R_{20} = 39.2\Omega$ e $R_{21} = 1k7$

$$A_v = 44.36$$

A resistência Rg foi dimensionada na ordem dos 39Ω , uma vez que segundo Johnson noise para este valor de resistência o ruído gerado pelas resistências não tem influência no ruído da saída [34].

Incorporou-se um relay non latching, 1 coil, para alta frequência que coloca as linhas adaptadas. Este comuta entre o sinal proveniente do filtro com frequência de corte de 166MHz e o sinal proveniente do amplificador. Para comutar este relay é utilizado um pino do microcontrolador auxiliado de um mosfet VN2222, uma vez que quando o pino do microcontrolador se encontra em high a tensão deste não é sufuciente para fazer comutar o relay. Com o recurso ao mosfet, quando o pino do microcontrolador se encontrar a high este irá conduzir colocando um dos pinos do relay com a tensão de referência, fazendo comutar o relay.

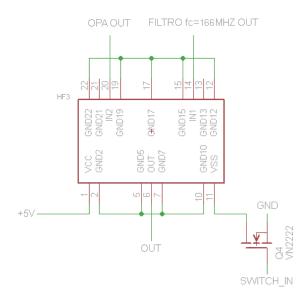


Figura 3.26: Configuração do Relay

O sinal proveniente do *relay* atravessa por um atenuador digital DAT-31R5-PP da MiniCircuits [35], programado pelo microcontrolador, que permite uma atenuação de 31.5dBV com uma escala de 0.5dBV. Com a utilização deste componente conseguese obter na saída um sinal com a amplitude programável de -64dBV até 0.5dBV.

Optou-se por este atenuador dado que este tem uma escala possui uma escala de 0.5dBV de atenuação, ser programado digitalmente e por possuir uma boa percisão de 0.1dBV, figura 3.27.

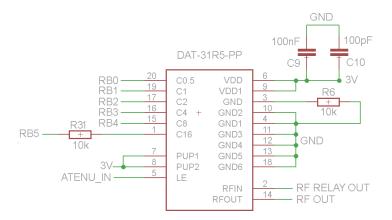


Figura 3.27: Configuração atenuador digital

Este atenuador é programado através de um microcontrolador, efectuada de forma paralela, como mostra a figura 3.28.

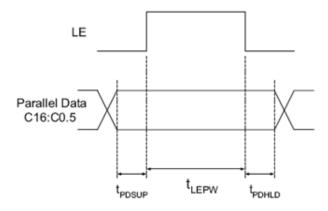


Figura 3.28: Diagrama temporal de transferência para o atenuador digital

Inicialmente coloca-se o bit de controlo LE a high e configura-se os restantes bits a high ou a low de acordo com a atenuação que se pretende obter e por fim volta-se a colocar o sinal de controlo a low. Este atenuador foi também configurado para que ao ligar o circuito o atenuador esteja automaticamente configurado para a atenuação máxima para protecção do circuito [35].

3.1.11 LCD

Também foi incorporada uma ligação a um lcd para permitir o envio de várias informações ao utilizador, permitindo futuramente a possível dispensa do PC, tornando este sistema portátil. O lcd utilizado foi um HJD204A baseado com o controlador Hitachi HD44780. A configuração utilizada encontra-se na figura 3.29 (programado apenas com 4 bits de dados).

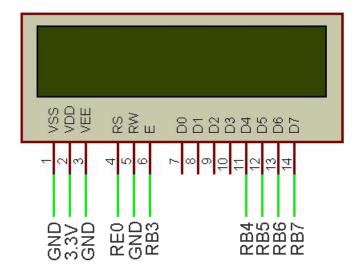


Figura 3.29: Configuração utilizada para o LCD

3.2 Pré Amplificador de entrada

Este circuito, figura 3.30, tem como função amplificar o sinal proveniente do meio de comunicação. Este circuito é constituído por 2 opamps na configuração não inversora (opa847 [34]), um relay e um atenuador digital variável (DAT-31R5-PP da MiniCircuits [35]) que permite o controlo digital do sinal de entrada, variando em torno deste desde os 0dBV a +60dBV com um passo de 0.5dBV, tudo controlado pelo microcontrolador da motherboard.

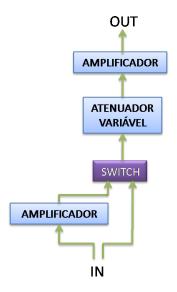


Figura 3.30: Diagrama de blocos da placa pré amplificador de entrada

A máxima amplitude do sinal de entrada neste circuito é de +11 dBV, limite imposto pelo atenuador digital variável.

O sinal de entrada atravessa por um opamp, OPA847, que se encontra na configuração não inversora, com um ganho de 35dBV, já utilizado no circuito anterior.

À saída deste amplificador encontra-se uma resistência em série de 27Ω, que tem como objectivo remover a carga parasita capacitiva melhorando a sua estabilidade. O sinal de saída do amplificador atravessa por um relay que comuta entre este sinal e o sinal de entrada controlado pelo microcontrolador. O sinal proveniente do relay atravessa por um atenuador digital (DAT-31R5-PP da MiniCircuits [35]) que permite uma atenuação máxima de 31.5dBV com uma escala de 0.5dBV, igualmente ao utilizado no circuito anterior.

Por fim encontra-se mais um andar de ganho igual ao referido anteriormente, constituído por um opamp com um ganho de 35dBV. Inicialmente foi referido que este circuito pode amplificar um sinal até +30dBV da sua amplitude inicial, contudo com a análise do ganho dos dois amplificadores poderá constatar-se à primeira vista que o ganho deste circuito seria superior ao mencionado. Todavia o facto para os amplificadores terem um ganho superior tem como objectivo efectuar uma compensação, uma vez que o relay introduz uma ligeira atenuação, mas principalmente devido à atenuação provocada pela resistência de saída do opamp.

3.3 Placa de Alimentação

Projectou-se uma fonte de alimentação, figura 3.31, com o objectivo de alimentar todos os circuitos existentes. Assim esta fonte fornece a tensão de saída de +5V, +3V, +3.069V, +3V, +1.8 e -5V. Na figura 3.31 encontra-se o diagrama de blocos da placa de alimentação projectada.

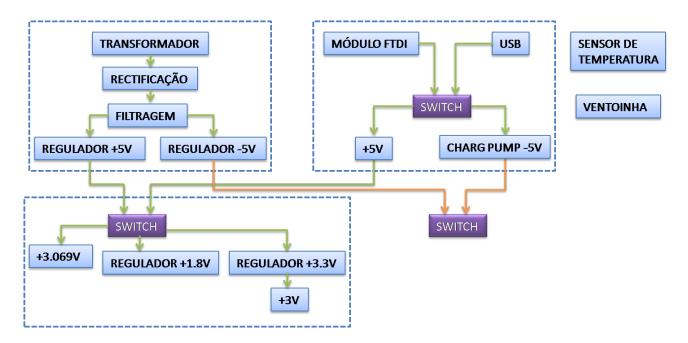


Figura 3.31: Diagrama de blocos da placa de alimentação

3.3.1 Tensão de +5V e -5V

Esta fonte de alimentação tem a versatilidade de se poder optar pela alimentação através do fornecimento de energia pela rede ou por uma interface usb de um computador. Para seleccionar o método de alimentação que se pretende utilizar basta usar um jumper.

3.3.1.1 Alimentação através da rede

A alimentação através da rede é obtida através do uso de um transformador de 2x6VA. O alimentação proveniente do transformador é subdividido em duas partes, uma para obtenção de tensão positiva, outra para a negativa. Em ambas as partes o alimentação atravessa por um rectificador de onda e por um condensador de filtragem. De seguida na parte positiva o sinal atravessa pelo regulador de tensão LM7805 que regula a tensão para +5V e permite um débito no máximo de 1A [36]. Na parte negativa o sinal atravessa pelo regulador de tensão LM7912 que regula para -12V e de seguida o sinal atravessa pelo regulador LM7905 que regula para -5V. O regulador LM7912 e o LM7905 permitem um débito máximo de 1.5A [37]. A configuração utilizada pode ser encontrada na figura 3.33.

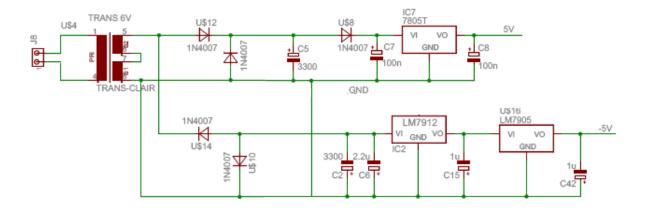


Figura 3.32: Configuração utilizada para alimentação através da rede

3.3.1.2 Alimentação através do USB

Quando o circuito é alimentado a partir da interface usb, obtêm-se directamente +5V, uma vez que é esta tensão que o usb fornece. Para obter -5V utilizou-se um charge pump (max889) que converte a tensão de +5V proveniente do usb para -5V. Este charge pump consegue fornecer à saída uma corrente máxima de 200mA [38]. A alimentação obtida pela interface usb pode ser adquirida por duas formas, através de uma ligação directa à interface usb ou através de um módulo FTDI [39] de conversão de usb para RS232. A ligação através da interface usb consegue debitar no máximo uma corrente de 500mA caso esteja ligado a portas usb2.0 [40]. A configuração utilizada pode ser encontrada na figura 3.33.

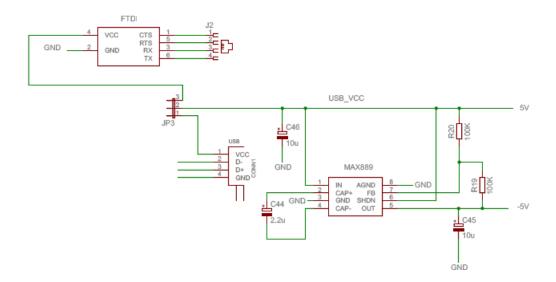


Figura 3.33: Configuração utilizada para para a alimentação através do USB

3.3.2 Tensão de 3.3V, 3V, 3.069 e 1.8V

Estas tensões simétricas de +5V e -5V têm como finalidade alimentarem vários opamps e permitir também a obtenção de outros valores de tensões. Com o recurso ao regulador tl431 consegue-se obter uma tensão de referência de +3.069V. A corrente que o tl431 consegue debitar é pequena, mas como a aplicação desta tensão é para referência da ADC do microcontrolador, não existe problema relativamente a esse ponto [41].

A partir desta tensão recorrendo a um divisor resistivo e a um *buffer* de tensão obtém-se também uma tensão de referência de 1.8V que tem também como finalidade o uso de referência para a ADC do microcontrolador.

Com o uso do regulador NJM2845DL consegue obter á saída 1.8V a partir de +5V, conseguindo debitar 800mA. Utilizaram-se dois reguladores iguais uma vez que o DDS tinha o requisito de duas fontes de tensão de 1.8V independentes [42].

Utilizou-se o regulador LM2937ET-3.3 para obter na saída uma tensão de +3.3V a partir de +5V de forma a poder alimentar o microcontrolador, o MAX232 e outros dispositivos. A corrente máxima que este pode fornecer é de 500mA [43]. A partir desta tensão com o auxílio de um díodo de schottky BAT46 obtêm-se a tensão de +3V que tem como finalidade de alimentar um relay.

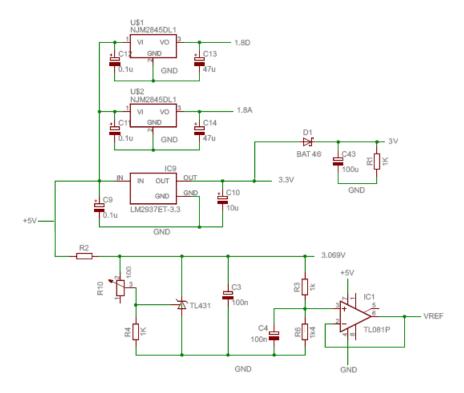


Figura 3.34: Configuração utilizada para obter 3.3V, 3V, 3.069V e 1.8V

3.3.3 Sensor de temperatura e FAN

Implementou-se neste circuito um sensor de temperatura, lm335, que tem uma exactidão de 1ºC com o intuito de medir a temperatura ambiente [44]. Adicionou-se também uma ventoinha com controlo ON/OFF. Nesta configuração colocou-se um díodo de roda livre para proteger o transístor VN2222. A configuração utilizada para o sensor de temperatura e para a ventoinha encontra-se na figura 3.35.

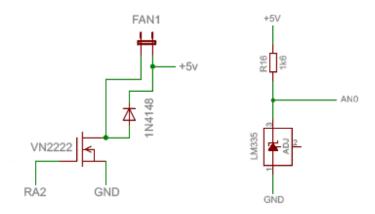


Figura 3.35: Configuração utilizada para o sensor de temperatura e para a ventoinha

Para protecção do circuito do utilizador colocou-se um fusível na entrada da ligação à rede e um interruptor.

Capítulo 4

Software

Neste capítulo irá ser feita uma descrição detalhada de todo o software desenvolvido, quer a nível do microcontrolador, quer da interface gráfica no computador pessoal. De uma forma geral, inicialmente os dados são introduzidos na aplicação gráfica e transferidos então do PC para o microcontrolador. Com os dados recebidos são calculada(s) a(s) frequência que se pretende analisar. Posteriormente o microcontrolador reprograma o DDS para que este gere a frequência desejada. De seguida determina os vários parâmetros S para essa frequência, e envia imediatamente a seguir os resultados para o PC. Este processo é repetido até que todas as frequências sejam analisadas, figura 4.1.

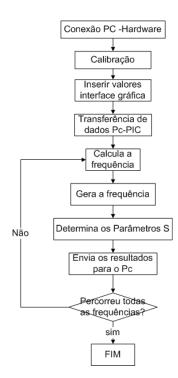


Figura 4.1: Diagrama geral de código do microcontrolador

4.1 Interface Gráfica

Para tornar a utilização do sistema desenvolvido mais simples e intuitiva, desenvolveuse uma interface user-friendly desenvolvida em "Visual Basic". Na figura 4.2 poderá ser visualizado o menu inicial da interface gráfica e as várias opções que o utilizador poderá selecionar.

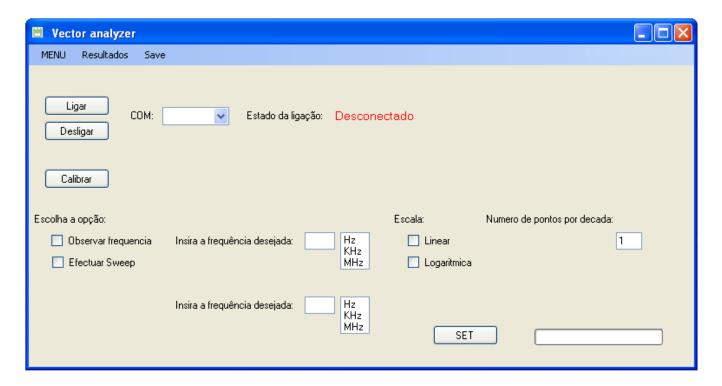


Figura 4.2: menu inicial da interface gráfica

Na utilização da interface, o primeiro passo é a conexão entre o PC e o micro-controlador. Para tal deverá escolher-se a porta COM com que se pretende conectar e pressionar o botão "conectar". Caso a conexão esteja correctamente efectuada no estado da ligação aparecerá a mensagem "Conectado", caso contrário continuará a mensagem "Desconectado".

Antes de efectuar qualquer estudo, o utilizador deverá calibrar o sistema. Para tal deverá ligar previamente a placa "pré amplificador de entrada" e "motherboard" utilizando os dois cabos fornecidos para esse efeito e de seguida o utilizador deverá escolher a opção "calibrar".

De seguida poderá escolher-se 2 opções, ou efectuar o estudo a apenas uma frequência ou realizar um sweep na frequência. Caso se pretenda efectuar um sweep na frequência, terá de se escolher o tipo de escala (linear ou logaritmica) e o número de pontos por década pretendidos.

Quando todos os campos estiverem preenchidos, o pedido é enviado para o microcontrolador após se pressionar o bottão "SET".

Para visualizar os resultados obtidos, basta pressionar na barra de menu a opção "resultados". Quando a opção pretendida for efectuar um sweep na frequência surgiram oito gráficos, com a resposta em frequência em amplitude e fase dos vários parâmetros S, figura 4.3.

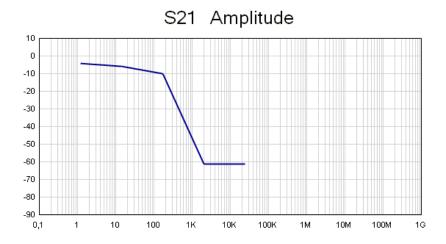


Figura 4.3: Gráfico com a resposta em frequência da amplitude do parâmetro S_{21}

Também é possível guardar os resultados para futura análise e reflexão, quer em formato de texto, quer em formato gráfico, para tal deverá ser pressionada a opção "Save" na barra de menu, figura 4.4.

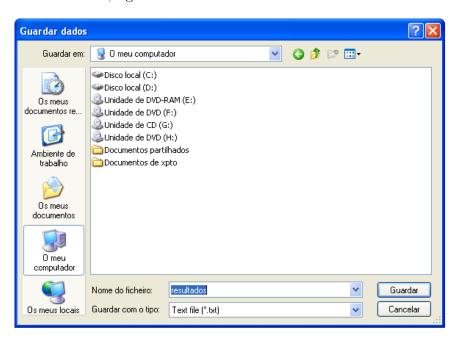


Figura 4.4: Menu para guardar dados

4.1.1 Tramas de envio

Este programa interage com o microcontrolador através da interface RS232 onde a troca de dados é efectuada da forma como mostra a figura 4.5. O microcontrolador usado é um PIC18LF458 da microchip [16]. Daqui para a frente será designado simplesmente por PIC.

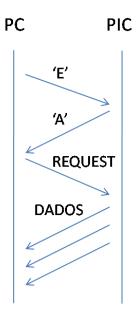


Figura 4.5: Fluxograma temporal da comunicação entre o o PC e o PIC

Para iniciar a transferência de dados o PC envia o caracter 'E' e caso a ligação esteja bem efectuada, o PIC responde com um acknowledge, caracter 'A'. De seguida o PC envia para o PIC um request indicando quais as frequências para analisar a resposta do DUT. Por fim, o PIC envia para o PC os resultados obtidos para cada frequência solicitada. A trama com o pedido pode ser constituída de 2 formas, caso pretenda-se analisar a resposta de uma única frequência, ou de várias, figuras 4.6 e 4.7.

CABEÇALHO	FREQUÊNCIA 1	GRANDEZA 1	FREQUÊNCIA 2	GRANDEZA 2	TIPO DE SWEEP	Nº AMOSTRAS	EOF
1 byte	5 bytes	1 byte	5 bytes	1 byte	1 byte	4 bytes	1 byte

Figura 4.6: Trama de envio para analisar várias frequências

1 byte 5 bytes 1 byte 1 byte	1 .	FREQUÊNCIA 5 bytes	GRANDEZA 1 byte	EOF 1 byte
------------------------------------	-----	-----------------------	--------------------	---------------

Figura 4.7: Trama de envio para analisar uma única frequência

Se se pretender observar uma determinada gama de frequências o PC envia uma trama constituída por um cabeçalho, o valor numérico da frequência pelo qual se pretende iniciar o sweep seguida da sua ordem de grandeza. Posteriormente segue-se o valor número da frequência e a sua respectiva ordem de grandeza do limite do sweep. Por fim é enviado o tipo de varrimento pretendido: se de forma linear, se logaritmica, o número de pontos desejado por década e um indicador fim de transmissão. Caso se pretenda observar a resposta a uma única frequência a trama enviada é constituída por um cabeçalho, o valor numérico da frequência seguida da sua ordem de grandeza e por um indicador de fim de transmissão.

Ao pedido efectuado, por cada frequência o PIC envia a trama que se encontra na figura 4.8. Esta é constituída por um cabeçalho, o valor numérico da frequência em estudo a respectiva ordem de grandeza, a amplitude e fase e um indicador de fim de transmissão.

CABEÇALHO FREQUÊNCIA GRANDEZA FASE S21 AMPLITUDE S21 1 byte 5 bytes 1 byte 3 byte 6 bytes	EOF 1byte
---	--------------

Figura 4.8: Trama enviada pelo PIC

Através da análise da figura 4.9 poderá-se observar dois fluxogramas temporais. Na situação a), o PIC responde ao pedido dentro do timeout imposto pelo PC, enquanto que na situação b), o PIC não responde dentro do tempo previsto. Neste caso, o PC efectua mais 2 tentativas de comunicação e caso em nenhuma delas o PC receba a confirmação, o utilizador é informado da ocorrência de uma falha na comunicação.

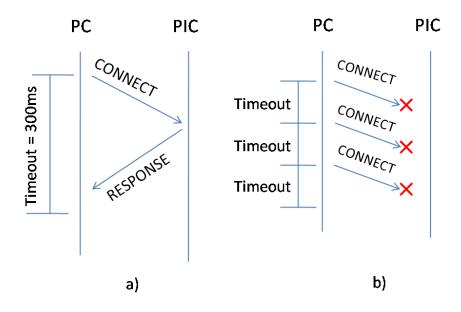


Figura 4.9: Fluxograma Temporal

4.2 Microcontrolador

4.2.1 Calibração

O factor de calibração é determinado no PC devido ao menor recurso do PIC. A calibração é automática e feita sempre antes de se efectuar qualquer medição, desta forma consegue-se eliminar o efeito de "envelhecimento" dos componentes e possíveis alterações ambientais, como por exemplo a temperatura.

Para efectuar a calibração teve-se em conta os vários "caminhos" que o sinal poderia percorrer, dependendo da posição dos relays e do multiplex, 4 combinações devido aos relays e 4 devido aos 4 filtros existentes, o que perfaz num total de 16 combinações possíveis. Assim sendo, para determinar os factores de compensação para cada combinação, decidiu-se por analisar a resposta de 20 frequências. Para cada filtro são analisadas 5 frequências e para cada frequência é analisada a resposta aos 4 caminhos definidos pelos relays. Em cada análise são efectuadas várias medições, efectuando de seguida uma média dos valores lidos, de forma a remover alguma oscilação existente.

As frequêcias escolhidas foram selecionadas tendo em conta a frequência de trabalho de cada filtro e de forma a ficarem linearmente espaçadas, sendo contempladas as frequências de transição entre filtros.

Para efectuar a calibração da fase é analisada simplesmente a resposta a uma dada frequência.

O diagrama de blocos do código da calibração encontra-se na figura 4.10.

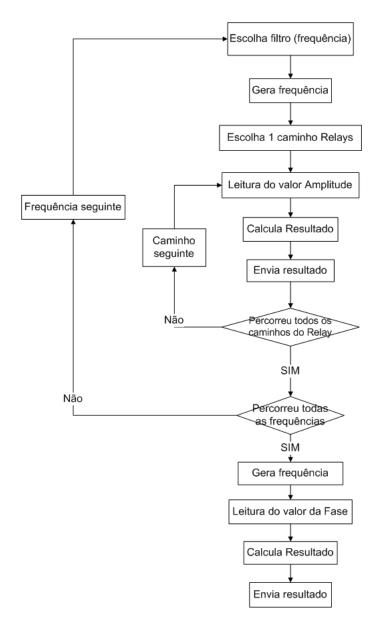


Figura 4.10: diagrama de blocos do código da calibração

4.2.2 Cálculo das frequências

Quando se pretende efectuar um varrimento na frequência, por parte do utilizador é apenas indicado o número de pontos por década e a frequência inicial e final do varrimento, tornando necessário o cálculo das frequências que se pretendem analisar por parte do microcontrolador.

Antes do cálculo propriamente dito, é efectuado um ajuste aos limites do varrimento, ou seja, não seria muito "conveniente" começar a efectuar o varrimento a partir de uma dada frequência independentemente do seu valor. Assim sendo, o mais razoável seria alargar a gama pretendida para que o varrimento começasse no início da década

imediatamente anterior à frequência inicial e terminasse na década imediatamente a seguir à frequência final.

De seguida, já com os novos limites, as frequências de interesse são calculadas. Para um varrimento cujo tipo de escala seja logaritmica é utilizado um algoritmo cuja representação em diagrama de blocos se encontra na figura 4.11.

Este algoritmo baseia-se no cálculo linear entre duas décadas sucessivas, ou seja, é calculado a distância entre frequências, passo, para que numa década exista o número de frequências pretendidas. Desta forma, todas as frequências cujo valor seja o valor inicial da década em estudo acrescentado de um valor múltiplo do passo são as frequências de interesse. Este procedimento é repetido para todas as décadas em análise.

Inicialmente o microcontrolador calcula a distância entre frequências e a frequência da próxima década. De seguida determina todas as frequências de interesse nessa década. Depois o valor da frequência toma o valor da frequência da década, uma vez que como o número de casas decimais é limitado a última frequência calculada não seria exactamente o valor da frequência da década mas sim um valor muito próximo devido aos arredondamentos que foram acumulados.

Como o formato do valor da frequência é composta por uma parte numérica e pela sua grandeza, o PIC verifica se o valor da década é igual a 1000 e caso seja, o valor da década passa a ser unitário e a sua grandeza é aumentada.

Caso a grandeza da frequência final seja superior à grandeza da última frequência o ciclo é repetido, caso contrário, se não existir uma década entre a última frequência e a frequência final o cálculo das frequências termina, senão o ciclo repete-se.

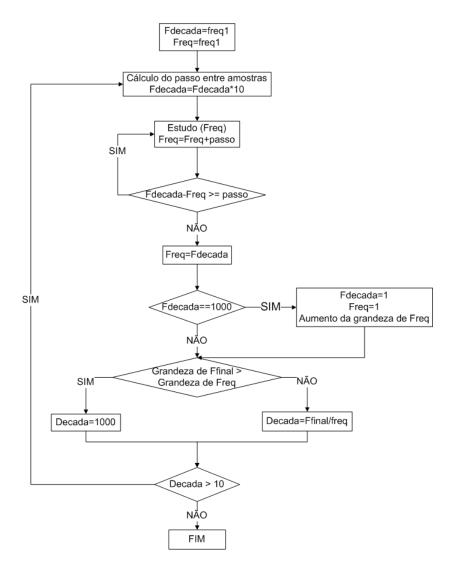


Figura 4.11: Diagrama de blocos do código para calcular as frequências na escala logaritmica

Para um varrimento em que a escala pretendida seja linear, o algoritmo utilizado baseia-se no cálculo do passo desde a frequência inicial e final e a soma de múltiplos deste à frequência inicial. Todas as frequências cujo valor seja o valor da frequência inicial acrescentado de um valor múltiplo do passo são frequências de interesse. O diagrama de blocos do algoritmo utilizado encontra-se na figura 4.12.

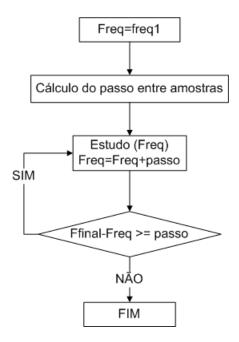


Figura 4.12: Diagrama de blocos do código para calcular as frequências na escala linear

4.2.3 Geração de frequências

Após determinar uma dada frequência é necessário programar o DDS para que esta a possa gerar, para tal o microcontrolador utiliza a interface SPI.

Inicialmente o *DDS* é configurado; para tal, em primeiro lugar é necessário efectuar reset para garantir que se encontre no estado inicial, de seguida são programados dois registos de configuração e por fim é efectuado o update (transição de '1' para '0' de um pino de controlo) para as novas configurações sejam aplicadas.

Após estar configurado, para gerar uma dada frequência basta configurar o registo relativamente à geração de frequências (FTW), ver datasheet [14], e efectuar um update para que as novas configurações se reflictam, como mostra a diagrama de blocos da figura 4.13. O valor do registo FTW é obtido aplicando a seguinte fórmula: $FTW = \frac{2^{32} \times Freq}{400 \times 10^6}$. A nível de software decidiu-se por subdividir o cálculo do registo por ordem de grandeza da frequência, para que ao efectuar o cálculo no microcontrolador não se perdesse resolução.

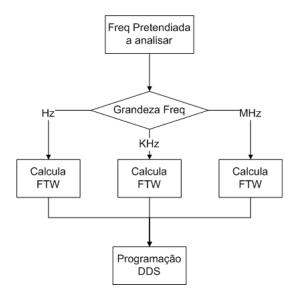


Figura 4.13: Diagrama de blocos da geração de uma dada frequência

4.2.4 Determinação dos parâmetros S

Após a frequência que se pretende analisar esteja gerada é verificada qual a variação quer em termos de amplitude e de fase do sinal. Para tal procedeu-se da seguinte forma como demonstra o diagrama de blocos da figura 4.14.

Inicialmente os atenuadores digitais programáveis estão configurados para a atenuação máxima e os opamps não estão seleccionados, para que não exista a possibilidade do sinal proveniente do DUT seja demasiadamente elevado, o que poderia danificar o detector de sinal. Assim sendo, no inicio o sinal recebido terá a maior atenuação possível.

De seguida é efectuada uma leitura da ADC do microcontrolador e caso o valor lido seja nulo aumenta-se a excitação. A excitação é aumentada diminuindo gradualmente o valor da atenuação provocada pelo sistema global (atenuadores digitais e opamps). Este processo é repetido até ser detectado algum valor na ADC do microcontrolador.

Todo o processo mencionado até então, é repetido para os parâmetros S_{11} e S_{21} .

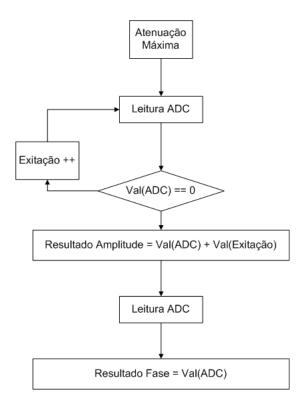


Figura 4.14: Diagrama de blocos do código para a determinação dos parâmetros S

O método acima referido só é válido para os parâmetros S_{11} e S_{21} , uma vez que para os outros dois parâmetros o sinal de referência não é o sinal emitido mas sim o sinal reflectido da carga, ou seja para calcular os outros dois parâmetros é necessário recorrer a outra forma.

Uma solução possível passaria inicialmente por calcular os parâmetros S_{11} e S_{21} , ficando a saber-se então à partida qual o "ganho para a frente". Através do opamp e do atenuador digital programável da placa motherboard colocaria-se o sinal de excitação com uma dada amplitude para que com o "ganho para a frente" do DUT, o sinal ficasse com a amplitude que permitisse a máxima gama dinâmica de detecção dos detectores de sinal, ou seja, -43dBV. A seguir a esta fase procederia-se simplemeste à atenua-ção/amplificação do outro sinal para comparação. De referir que a gama dinâmica total para os parâmetros S_{12} e S_{22} é inferior que para os outros dois parâmetros, uma vez que apenas poderia-se condicionar o sinal com os opamps e atenuadores da placa pré amplificador de entrada.

Capítulo 5

Resultados

Neste capítulo irão ser apresentados os resultados do trabalho realizado durante o projecto. Nas figuras que se seguem estão representadas as pcbs das várias placas desenvolvidas, bem como a forma de onda do sinal em determinados pontos de maior relevância.

5.1 Pcb's

Nas figuras que se seguem, 5.2, 5.1, 5.3 e 5.4 podem ser observadas as várias pcbs realizadas para a construção do *setup* final.

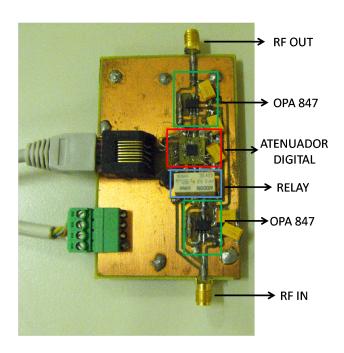


Figura 5.1: Pcb da placa pré amplificador de entrada

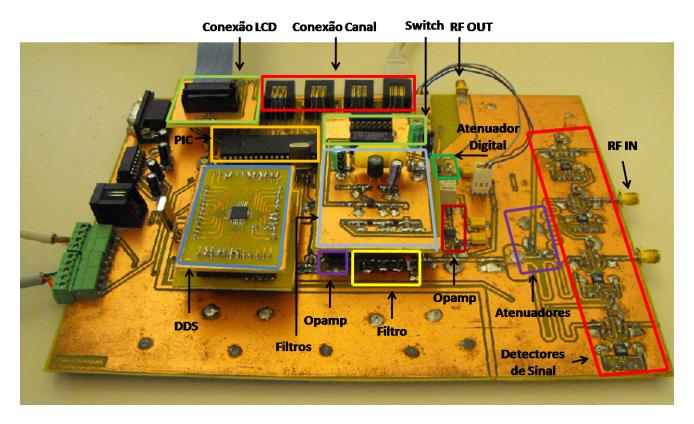


Figura 5.2: Pcb da placa Motherboard



Figura 5.3: Pcb para conexão do LCD $\,$

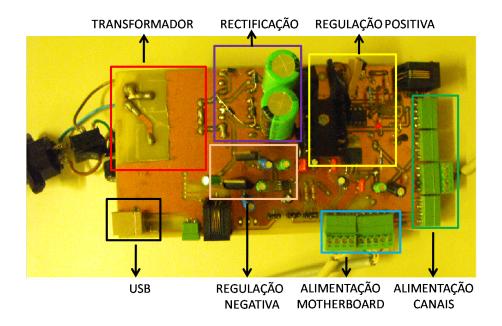


Figura 5.4: Pcb da Placa de Alimentação

5.2 Formas de onda do sinal gerado

Nesta secção irão ser mostradas as formas de onda do sinal produzido em vários pontos de interesse, tais como, o sinal à saída do DDS, à saída do opamp em configuração diferença, à saída do filtro e o sinal à entrada do detector de sinal.

5.2.1 Formas de onda à saída do DDS

Nas várias imagens que se seguem poderá verificar a forma de onda do sinal gerado pelo *DDS* para várias frequências. Poderá-se verificar que o sinal se encontra sem ruído e que a amplitude do sinal mantém constante independentemente da frequência do sinal.

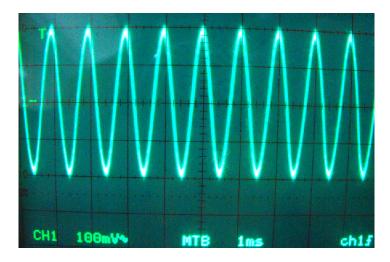


Figura 5.5: Forma de onda com frequência de 1KHz à saída do DDS

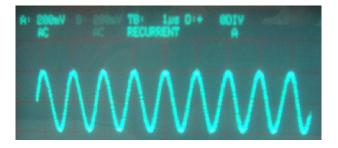


Figura 5.6: Forma de onda com frequência de 1MHz à saída do DDS

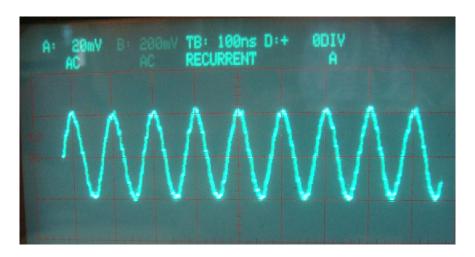


Figura 5.7: Forma de onda com frequência de $10\mathrm{MHz}$ à saída do DDS

5.2.2 Formas de onda à saída do Opamp configuração diferença

Através da observação da imagem que segue pode-se constatar que o sinal à saída do Opamp na configuração diferença se encontra limpido, sem ruído e com a amplitude

tal como esperado.

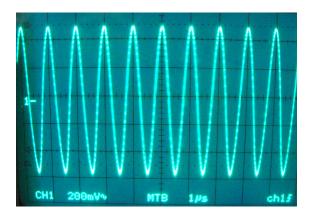


Figura 5.8: Forma de onda com frequência de 1MHz à saída do opamp

5.2.3 Formas de onda à saída do Filtro fc=166MHz

Através da observação da imagem que se segue pode-se também comprovar que o sinal à saída do filtro se encontra como previsto com uma amplitude de 135mVp.

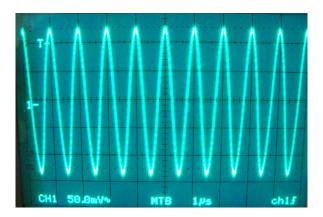


Figura 5.9: Forma de onda com frequência de 1MHz à saída do filtro fc=166MHz

5.2.4 Formas de onda à entrada do detector de sinal

Na figura que se segue encontra-se representado o sinal de referência para o detector de sinal. Como se pode verificar o sinal têm uma amplitude de 10mVp tal como esperado. O facto de o sinal não se apresentar "bem definido" deve-se ao osciloscópio ter dificuldade em efectuar o *triger*, uma vez que este sinal possui uma amplitude baixa.

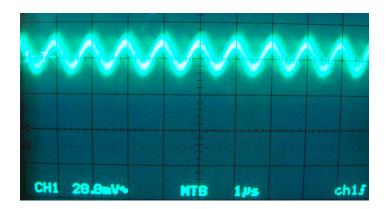


Figura 5.10: Forma de onda com frequência de 1MHz à entrada do detector de sinal

5.3 Resposta do detector de sinal

No gráfico que se segue, figura 5.11, poderá verificar-se que o detector de sinal possui uma relação de 23 mV/dB e não o inicialmente previsto de 30 mV/dB. Para efectuar este teste utilizou-se um sinal sinusoidal, onde se fez variar a sua amplitude em passos de 0.5 dB recorrendo para tal ao atenuador digital programável.

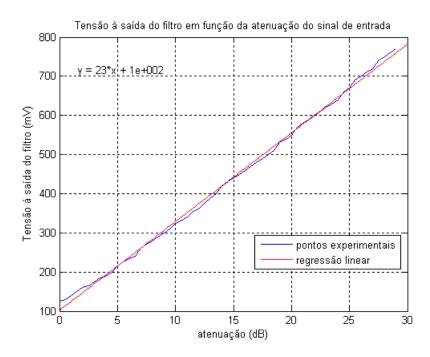


Figura 5.11: Gráfico da tensão à saída do filtro em função da atenuação do sinal de entrada

75

5.4 Calibração

Para efectuar a calibração conectou-se a placa pré amplificador de entrada directamente à placa motherboard, desta forma o DUT é considerado como um curto-circuito. Aos valores obtidos nesta situação são determinados os factores de compensação para que os resultados obtidos sejam considerados como nulos, ou seja, 0dB para a amplitude e 0 graus de diferença de fase, para todas as combinações possíveis por onde o sinal atravesse.

5.4.1 Amplitude

Na calibração da amplitude, ao analisar a resposta obtida das várias frequências constatou-se que a variação de amplitude para cada uma das 16 combinações em que o sinal poderia atravessar não excede 0.3 dB em torno do seu valor médio, figura 5.12.

Os valores recebidos, reflectem além da diferença de amplitude entre o sinal emitido e recebido, reflectem também a queda de tensão do sinal DC proveniente do detector de sinal até à entrada da ADC do microcontrolador, provocada pela passagem do sinal pelos multiplexers e pelos filtros.

Ao efectuar uma medição, ao valor lido será adicionado o factor de compensação de forma a compensar alguma variação na resposta dos vários dispositivos electrónicos e a correcção do valor lido perante a atenuação do sinal ao atravessar pelos filtros e pelos multiplexers.

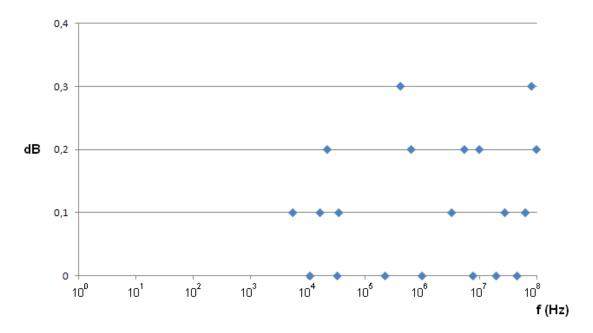


Figura 5.12: Resultados Obtidos

5.4.2 Fase

Para poder calibrar a medição da fase tem de se garantir que os "cabos" utilizados para interligar a placa motherboard e cada canal sejam sempre os mesmos quando se estiver a calibrar e quando se estiver a utilizar o sistema. Caso não se proceda desta forma ao efectuar testes, existirá um erro na medição.

Com os cabos utilizados, ao efectuar a calibração verificou-se que a diferença de fase existente era de 4 graus, ou seja, como a distância percorrida pelos dois sinais (de referência e do proveniente do DUT) até ao detector não é a mesma, existirá uma diferença de fase entre os dois sinais. Esta diferença de fase dependerá do tamanho dos "cabos" utilizados.

Neste caso, a gama de diferença de fase que o detector irá detectar não será de 0 a 180 graus, mas sim de -4 a 176 graus, ou seja, existirá uma deslocação na gama de detecção.

Capítulo 6

Conclusões e Trabalho Futuro

O objectivo deste projecto teve como base a construção de um setup que podesse ser portátil, fácil de transportar, que permitisse efectuar a caracterização de propagação de sinais pelo corpo humano. O objectivo proposto inicialmente foi construir um sistema que permitise determinar apenas o "ganho para a frente", mais conhecido pelo parâmetro S_{21} .

Neste trabalho foi-se mais além do proposto, além da determinação apenas do valor absoluto também é possível determinar a fase. Outro aspecto de realçar é o facto de além se determinar o parâmetro S_{21} o sistema foi projectado com o intuíto de permitir a detecção dos restantes 3 parâmetros, efectuando desta forma uma caracaterização completa do DUT. Embora o sistema esteja dimensionado para os 4 canais, a nível de software apenas teve-se em conta 1 canal, dada as limitações de tempo.

Um dos problemas encontrados durante a execução do trabalho residiu na impedância caracteristica da linha, o que fez que tivesse de alterar tanto o atenuador inicialmente projectado, como o filtro do DDS. Estava inicialmente prevista uma impedância de linha 50Ω , para uma PCB em FR4 (1.6mm de espessura) obrigava a uma largura de pista demasiadamente grande.

Reflectindo sobre o trabalho pode-se constatar que os objectivos propostos foram cumpridos excepto a não apresentação de resultados para um varimento na frequência, embora esteja implementado a nivel de *software*. Relativamente ao sinal sinusoidal gerado verifica-se que se apresenta límpido, sem ruído em qualquer parte do circuito com qualquer frequência.

Não foi possível realizar os "testes finais" no qual visava verificar a resposta em frequência de vários DUT e comparar com a resposta obtida com a de um network analyzer, dado que o detector de sinal avariou após realizar os testes da calibração.

Relativamente ao custo do sistema desenvolvido, este é bastante mais económico

que um vector analyzer comercial. Como é obvio este apresenta diversas limitações em comparação com um comercial, mas inclui algumas particularidades vantajosas para o estudo de propagação de sinais pelo corpo humano, sendo de destacar o possuir isolamento óptico na ligação ao DUT. Um sistema comercial ronda no mínimo por volta dos $20.000\mathfrak{C}$, enquanto que para este sistema define-se um custo aproximado de $300\mathfrak{C}$, excluíndo gastos de mão de obra:

- Motherboard: 170€

- 4 canais: 80€

- Placa de alimentação: 40€

- Diversos: 10€

Relativamente a trabalho futuro com vista a melhorar o sistema, um dos aspectos em ter em conta seria elaborar um novo *hardware*, onde se possa assemblar os componentes de menor dimensão directamente na placa e não efectuar sub-assemblagem como o realizado. Outro aspecto de referir seria a utilização de *power splitter* na saída do filtro do *DDS* de forma a garantir a mesma impedância de saída em cada uma das saídas do filtro.

A nível de *software* seria conveniente maximizar o nível de código da interface gráfica e relativamente ao código do microcontrolador implementar o código para determinar os valores dos restantes 2 parâmetros S.

Por fim realizar um varimento na frequência de vários DUT e comparar com os resultados obtidos com os obtidos por um $vector\ analyzer$.

Bibliografia

- [1] Yong Pei Bin Wang. Body area networks.
- [2] http://revistareclame.com.br/wp-content/uploads/2008/09/info_bluetooth.jpg consultado pela última vez em 15 de Julho de 2009.
- [3] http://informatica.hsw.uol.com.br/rede-wifi.htm consultado pela última vez em 15 de Julho de 2009.
- [4] http://en.wikipedia.org/wiki/File:Eazix_numbered.jpg consultado pela última vez em 15 de Julho de 2009.
- [5] www.ieee802.org/802_tutorials/06-July/15-06-0331-00-0ban-tutorial-on-body-area-networks.ppt consultado pela última vez em 15 de Julho de 2009.
- [6] http://www.redtacton.com/en/info/index.html consultado pela última vez em 15 de Julho de 2009.
- [7] M. A. Hanson, H. C. Powell, A. T. Barth, K. Ringgenberg, B. H. Calhoun, J. H. Aylor, and J. Lach. Body area sensor networks: Challenges and opportunities. *Computer*, 42(1):58–65, Jan. 2009.
- [8] http://www.ist-runes.org/scenario.html consultado pela última vez em 15 de Julho de 2009.
- [9] http://www.engadget.com/2006/11/20/etris-body-area-network-prototype-prints-through-your-body/ consultado pela última vez em 15 de Julho de 2009.
- [10] Thomas Guthrie Zimmerman. Personal area networks (pan): Near-field intra-body communication. 1995.

80 BIBLIOGRAFIA

[11] K. Hachisuka, A. Nakata, T. Takeda, Y. Terauchi, K. Shiba, K. Sasaki, H. Hosaka, and K. Itao. Development and performance analysis of an intra-body communication device. In *Proc. TRANSDUCERS*, Solid-State Sensors, Actuators and Microsystems, 12th International Conference on, volume 2, pages 1722–1725, 8–12 June 2003.

- [12] J. A. Ruiz, Jiang Xu, and S. Shimamoto. Propagation characteristics of intrabody communications for body area networks. In *Proc. 3rd IEEE Consumer Communications and Networking Conference CCNC 2006*, volume 1, pages 509–513, 8–10 Jan. 2006.
- [13] http://www.analog.com/library/analogDialogue/archives/38-08/dds.html consultado pela última vez em 15 de Julho de 2009.
- [14] http://www.analog.com/static/imported-files/data_sheets/AD9859.pdf consultado pela última vez em 15 de Julho de 2009.
- [15] www.ieee.li/pdf/essay/dds.pdf consultado pela última vez em 15 de Julho de 2009.
- [16] http://ww1.microchip.com/downloads/en/DeviceDoc/spi.pdf consultado pela última vez em 15 de Julho de 2009.
- [17] http://cp.literature.agilent.com/litweb/pdf/5965-7917E.pdf consultado pela última vez em 15 de Julho de 2009.
- [18] http://na.tm.agilent.com/pna/files/eld.html consultado pela última vez em 15 de Julho de 2009.
- [19] www.analog.com/static/imported-files/data_sheets/AD8302.pdf consultado pela última vez em 15 de Julho de 2009.
- [20] Seong-Jun Song, Namjun Cho, and Hoi-Jun Yoo. A 0.2-mw 2-mb/s digital transceiver based on wideband signaling for human body communications. 42(9):2021–2033, Sept. 2007.
- [21] M. Shinagawa, M. Fukumoto, K. Ochiai, and H. Kyuragi. A near-field-sensing transceiver for intrabody communication based on the electrooptic effect. 53(6):1533-1538, Dec. 2004.
- [22] http://www.embeddedworld.com.br/produtos.asp?produto=5000 consultado pela última vez em 15 de Julho de 2009.

BIBLIOGRAFIA 81

[23] http://www.eletronica.org/modules.php?name=News&file=article&sid=40 - consultado pela última vez em 15 de Julho de 2009.

- [24] http://www.techpluto.com/wifi-info/ consultado pela última vez em 15 de Julho de 2009.
- [25] http://www.meshnetics.com/zigbee-faq/ consultado pela última vez em 15 de Julho de 2009.
- [26] http://www.zigbees.com/ consultado pela última vez em 15 de Julho de 2009.
- [27] http://www.ieee.li/pdf/viewgraphs/direct_digital_synthesis.pdf consultado pela última vez em 15 de Julho de 2009.
- [28] http://www.embedded.com/story/OEG20020124S0116 consultado pela última vez em 15 de Julho de 2009.
- [29] Yeou-Song Lee. Testing dynamic accuracy of vector network analyzers using the 40 ghz step attenuator. In *Proc. 65th ARFTG Conference Digest Spring 2005*, pages 149–158, 17 June 2005.
- [30] http://www.hparchive.com/Application_Notes/HP-AN-95-1.pdf consultado pela última vez em 15 de Julho de 2009.
- [31] http://www.microwaves101.com/encyclopedia/sparameters.cfm consultado pela última vez em 15 de Julho de 2009.
- [32] http://www.mwrf.com/Articles/Print.cfm?ArticleID=5512 consultado pela última vez em 15 de Julho de 2009.
- [33] http://www.analog.com/static/imported-files/tech_articles/585795575654495533 87221439863make_precise_base-station_power_measurements.pdf consultado pela última vez em 15 de Julho de 2009.
- [34] http://focus.ti.com/lit/ds/symlink/opa847.pdf consultado pela última vez em 15 de Julho de 2009.
- [35] http://doc.chipfind.ru/mini/dat31r5pp.htm consultado pela última vez em 15 de Julho de 2009.
- [36] http://www.datasheetcatalog.org/datasheet/fairchild/LM7805.pdf consultado pela última vez em 15 de Julho de 2009.

82 BIBLIOGRAFIA

[37] http://www.hep.upenn.edu/SNO/daq/parts/lm7915.pdf - consultado pela última vez em 15 de Julho de 2009.

- [38] http://pdfserv.maxim-ic.com/en/ds/MAX889.pdf consultado pela última vez em 15 de Julho de 2009.
- [39] http://www.ftdichip.com/ consultado pela última vez em 15 de Julho de 2009.
- [40] http://en.wikipedia.org/wiki/USB consultado pela última vez em 15 de Julho de 2009.
- [41] http://www.fairchildsemi.com/ds/TLem 15 de Julho de 2009.
- [42] http://pdf1.alldatasheet.net/datasheet-pdf/view/119439/NJRC/NJM2845DL-25.html consultado pela última vez em 15 de Julho de 2009.
- [43] http://www.datasheetcatalog.org/datasheet2/3/070ofguzw3qzfh5xleheot5055fy.pdf consultado pela última vez em 15 de Julho de 2009.
- [44] http://www.national.com/mpf/LM/LM335.html consultado pela última vez em 15 de Julho de 2009.

Apêndice A

Código PIC

A.1 Menu.c

```
#include <pic18.h>
#include "driver.h"
#define ON 1
#define OFF 0
#define CANAL1 1
#define CANAL2 2
#define CANAL3 3
#define CANAL4 4
#define S2 LATD7
#define S1 LATD6
#define S0 LATD5
#define amplitude 1
#define fase 0
#define directo 1
#define opamp 0
#define Relay LATD0
                        //1 directo da entrada
#define RelayIn LATA4
//define do dds
#define update LATC2
#define reset LATC1
#define CFR1 0
#define CFR2 1
#define FTW0 4
void main (void)
//variaveis para recepcao dos dados do po
float freq1=0.0;
char g1;
float freq2=0.0;
char g2;
char escala;
int amostras=0;
//variaveis para calcular as frequencias para efectuar o sweep
float fx=0.0;
char gx;
int decada=0;
float passo=0.0;
unsigned int i=1;
float nfreq=0.0;
float fim=0.0;
   TRISA=0x0B;
   TRISB=0x00;
   TRISC=0xD0;
   TRISD=0x00;
   TRISE0=0;
   TRISE1=0;
   TRISE2=0;
```

A.1. MENU.C 85

```
dds_setup();
   usart_config();
   config_ad8302();
   while (receive_char() != 'E'); //Para conexão com o Pc
    send_char('A');
   while(receive_char() != 'C'); //Espera até receber confirmação
                                  // para fazer a calibração
    calibracao();
do{
    pctopic(&freq1,&g1,&freq2,&g2,&escala,&amostras); //recepção dos dados do pc
    limites(&freq1, &g1, &freq2, &g2); //calculo dos limites do intervalo de sweep
   fx=freq1;
   gx=g1;
   fim=freq1;
  if(amostras != 1) {
    do{
    fim=10*fim;
   passo=(fim-fx)/(amostras-1);
   while((fim-fx) >= passo) {
         send_char('D');
     print_num(10, fx);
         send_char(gx);
      estudo(fx, gx);
      fx=fx+passo;
    fx=fim;
   if(fim==1000) {
     fim=1;
     fx=1;
     if(g2 > gx){
       if(gx=='H'){
              gx='K';
       else{
         gx='M';
        }
     }
    }
    if(g2 > gx){
     decada=(freq2/fx) + 1000;
   else{
      decada=freq2/fx;
    }while(decada >= 10);
```

```
print_float(fx);
if(fx<100.0){
    send_char('0');
}

print_num(10, fx);
send_char(gx);
estudo(fx, gx);
send_char('E');

}while(1);</pre>
```

A.2. DRIVER.C 87

A.2 Driver.c

```
#include <pic18.h>
#include "driver.h"
#define S2 LATD7
#define S1 LATD6
#define S0 LATD5
#define update LATC2
#define reset LATC1
#define CFR1 0
#define CFR2 1
#define FTW0 4
#define Relay LATD0
                     //1 directo da entrada
#define RelayIn LATA4 //
// Configuração da usart
void usart_config(void)
//TRISC: configuração do PORTC
  TRISC6 = 1; //RC6 funciona como TX
  TRISC7 = 1; //RC7 funciona como RX
//TXSTA: Transmissão e Control register
  TX9 = 0; //8-bit de transmissao
             //transmissao activa
  TXEN = 1;
            //modo assincrono
  SYNC = 0;
              //BR high speed
  BRGH = 1;
//RCSTA: Recepção e Control register
  RX9 = \frac{0}{3}; //8-bit de recepção
  CREN = 1; //Continuous Receive Enable
              //transmissao por porta serie
  SPEN = 1;
  SPBRG =21; //baudrate = Fosc/(16(x+1))
}
void send_str(const char *pnt)
   while (*pnt!=0)
       send_char(*pnt);
      pnt++;
   }
}
void send_char(char car)
   while (!TXIF); // Espera até poder enviar um caracter
   TXREG=car;
                //Coloca no reg TXREG o caracter apontado por car
}
```

```
char receive_char(void)
{
    while(!RCIF); // Espera até receber um caracter
return RCREG; // Devolve o caracter recebido via rs232
}
void eco(void)
{ char car;
   car=receive_char();
   send_char(car);
}
void put_str (unsigned char *str)
{
    unsigned char *s=str;
    while (*s!='\0')
    {
        send_char(*s);
        s++;
    }
}
unsigned char bin2asc(unsigned char num)
    num += '0';
    if(num > '9')
       num += 'A'-'9'-1;
    return num;
}
void print_num(unsigned char base, unsigned int val)
{
    unsigned char str[12];
    unsigned char i, digito;
    i = 0;
    do {
       digito = val % base;
       val /= base;
       str[i++] = bin2asc(digito);
    } while( val > 0 );
    while( i > 0 ) {
      while( TXIF == 0 );
      TXREG = str[--i];
    }
}
unsigned int asc2dec(char aux)
    aux=aux-'0';
    return aux;
}
```

A.2. DRIVER.C 89

```
unsigned int receive_int(void)
    unsigned int total, aux2;
   char aux;
   total=0;
   while(1)
        aux=receive_char();
        if (aux!='q')
            aux2=asc2dec(aux);
            total=10*total+aux2;
           print_num(10, aux2);
        }
        else
        {
           return total;
        }
    return 0;
}
void val_string(char *str, unsigned int val)
   unsigned int i=0;
    for(i=0; i<3; i++)
       str[i]=' ';
    }
    i=0;
   if ((val/100)!=0) {
    str[i]=(val/100)+'0';
   val=val-((str[i]-'0')*100);
   i++;
   if((val/10)!=0){
    str[i]=(val/10)+'0';
    val=val-((str[i]-'0')*10);
   i++;
    str[i]=val+'0';
    i++;
}
```

```
void
DelayMs (unsigned char cnt)
#if XTAL_FREQ <= 2MHZ
       DelayUs(996);
   } while (--cnt);
#endif
    XTAL_FREQ > 2MHZ
   unsigned char i;
   do {
      i = 4;
       do {
          DelayUs(250);
       } while(--i);
   } while (--cnt);
#endif
}
#define LCD_RS RE0 // Register select
#define LCD_EN RB3 // Enable
#define LCD_D4 RB4 // Data bits
#define LCD_D5 RB5 // Data bits
#define LCD_D6 RB6 // Data bits
#define LCD_D7 RB7 // Data bits
#define LCD_STROBE ((LCD_EN = 1),(LCD_EN=0))
// write a byte to the LCD in 4 bit mode
void lcd_write(unsigned char c)
   if(c & 0x80) LCD_D7=1; else LCD_D7=0;
   if(c & 0x40) LCD_D6=1; else LCD_D6=0;
   if(c & 0x20) LCD_D5=1; else LCD_D5=0;
   if(c & 0x10) LCD_D4=1; else LCD_D4=0;
   LCD_STROBE;
      DelayUs(40);
   if(c & 0x08) LCD_D7=1; else LCD_D7=0;
   if(c & 0x04) LCD_D6=1; else LCD_D6=0;
   if(c & 0x02) LCD_D5=1; else LCD_D5=0;
   if(c & 0x01) LCD_D4=1; else LCD_D4=0;
   LCD_STROBE;
   DelayUs(100);
```

```
}
// Clear and home the LCD
void
lcd_clear(void)
   LCD_RS = 0;
   lcd_write(0x1);
   DelayMs(2);
// write a string of chars to the LCD
void
lcd_puts(const char * s)
   LCD_RS = 1; // write characters
   while(*s) lcd_write(*s++);
}
// write one character to the LCD
lcd_putch(unsigned char c)
   LCD_RS = 1; // write characters
   DelayUs(10);
   lcd_write(c);
}
// Go to the specified position
void
lcd_goto(unsigned char pos)
   LCD_RS = 0;
   lcd_write(0x80 + pos);
}
// initialise the LCD - put into 4 bit mode
void
lcd_init(void)
   LCD_RS = 0; // write control bytes
    DelayMs(15);// power on delay
```

```
LCD_D4 = 1; // init!
   LCD_D5 = 1; //
   LCD_STROBE;
   DelayMs(5);
   LCD_STROBE; // init!
   DelayUs(100);
   LCD_STROBE; // init!
   DelayMs(5);
   LCD_D4 = 0; // set 4 bit mode
   LCD_STROBE;
   DelayUs(40);
   lcd_write(0x28);// 4 bit mode, 1/16 duty, 5x8 font, 2lines
   lcd_write(0x0C);// display on
   lcd_write(0x06);// entry mode advance cursor
   lcd_write(0x01);// clear display and reset cursor
void lcd_started(void)
   lcd_init();
   lcd_clear();
   lcd_cursor(0x00);
   DelayMs(15);
   lcd_goto(0x00 + 2);
   lcd_puts("PROGRAM STARTED");
   DelayMs(15);
}
void 1m335(void)
   unsigned int val=0;
   unsigned int temp_int, temp_frac;
   char str_int[4];
   char str_frac[2];
   unsigned int xpto=0;
   ADRESH=0;
   ADRESL=0;
   ADCON1=0x04; //just esquerda
   ADCON0=0x81;
   DelayMs(20);
               //tempo de aquisição
   GODONE=1; // Iniciar conversao -> GO/Done\=1
   while (GODONE); // Aguarda final da conversao
```

```
val=ADRESH ; // valor com 8bits de resolucão
   DelayMs(20); //tempo de aquisição
   xpto=(12.72*val) - 2731; // 3270/255
   temp_int= xpto/10;
   temp_frac= xpto-(temp_int*10);
    val_string(str_int, temp_int);
    val_string(str_frac, temp_frac);
   lcd_goto(0x14);
   lcd_puts("TEMP AMB:");
   DelayMs(15);
   lcd_goto(0x1E);
   lcd_puts(str_int);
   DelayMs(15);
   lcd_goto(0x20);
   lcd_puts(".");
   DelayMs(15);
   lcd_goto(0x21);
   lcd_putch(temp_frac + '0');
   DelayMs(15);
   lcd_goto(0x23);
   lcd_puts("GRAUS");
   DelayMs(15);
}
void config_ad8302(void)
{
  ADCON1=0x05;
                   //justificação à esquerda com 10bits
  ADCON0=0x89;
                 //tempo de aquisição
  DelayMs(20);
  DelayMs(20);
                   //tempo de aquisição
unsigned long ad8302(void)
{
unsigned long val;
    GODONE=1; // Iniciar conversao -> GO/Done\=1
   while (GODONE); // Aguarda final da conversao
```

```
val= (ADRESH << 2) + (ADRESL>>6); // valor com 10bits de resolução
   DelayMs(20); //tempo de aquisição
   DelayMs(20); //tempo de aquisição
return val;
void fan(char val)
    if (val) {
      RA2=1;
     }
    else{
          RA2=0;
}
void relay(char val)
    if (val) {
      LATD0=1; //directo da entrada
    else{
      LATD0=0; //atravessa pelo opamp
}
void multiplex(char canal, char medida)
    switch (canal) {
       case 1:
            S2=0;
            S1=0;
             break;
        case 2:
            S2=0;
            S1=1;
            break;
        case 3:
            S2=1;
            S1=0;
             break;
        case 4:
            S2=1;
             S1=1;
             break;
```

```
if (medida) {
    S0 = 0;
   else{
       S0=1;
}
void escolhafiltros(float freq, char gx)
              //Se a frequencia for menor que 33KHz
 if(gx=='H'){
   LATE2=1;
  LATE1=0;
 else{
   if (gx=='K') {
    if(freq <= 33.0) {     //Se a frequencia for menor que 33KHz</pre>
   LATE2=1;
     LATE1=0;
              //Se a frequencia for entre 33KHz e 1MHz
    else{
      LATE2=0;
      LATE1=1;
    }
   if(freq <= 19.8) { //Se a frequencia for entre 1MHz e 19.8MHz
    LATE1=0;
   else{
                 //Se a frequencia for maior que 19.8MHz
    LATE2=1;
       LATE1=1;
   }
   }
 }
}
void pctopic(float *freq1, char *g1, float *freq2, char *g2, char *escala, int *amostras)
char car;
  do{
  car=receive_char();
```

```
while(( car != 'F') && (car != 'S'));
send_char(car);
if (car== 'F') {
 car =receive_char();
 *freql=(car-'0');
 car =receive_char();
  while(( car != 'H') && (car != 'K') && (car != 'M')){
   if (car == '.') {
     car =receive_char();
     *freq1= *freq1 + ((car-'0')*0.1);
   else{
     *freql= (*freq1*10) + (car-'0');
   car =receive_char();
  }
  *amostras=1;
  *gl=car;
  receive_char();
else{
 car =receive_char();
 while(( car != 'H') && (car != 'K') && (car != 'M')){
  if (car == '.') {
     car =receive_char();
     *freql= *freq1 + ((car-'0')*0.1);
  car =receive_char();
   1
   else{
     *freql= (*freql*10) + (car-'0');
     car =receive_char();
   }
  *gl=car;
                             // byts da freq2
  car=receive_char();
  while(( car != 'H') && (car != 'K') && (car != 'M')){
   if (car == '.') {
     car =receive_char();
     *freq2= *freq2 + ((car-'0')*0.1);
  car =receive_char();
   else{
  *freq2= (*freq2*10) + (car-'0');
     car =receive_char();
  *g2=car;
                          //grandeza de freg2
  *escala =receive_char();
                             //escala
                         // byts do número de amostras
  car =receive_char();
  while (car != 'E') {
```

```
*amostras= (*amostras*10) + (car-'0');
      car =receive_char();
   }
void limites(float *freq1, char *g1, float *freq2, char *g2)
unsigned int converter=0;
   if (*freq1 < 10) {</pre>
       converter=*freq1;
       *freql=converter;
    }
   else{
     if (*freq1 < 100) {</pre>
       converter=*freq1;
       converter= (converter/10) *10;
       *freql=converter;
     }
     else{
           converter=*freq1;
       converter= (converter/100) *100;
       *freql=converter;
         }
   if (*freq2 < 10) {
         converter=freq2;
     if (*freq2 > converter) {
        converter=converter + 1;
     *freq2=converter;
    }
   else{
     if (*freq2 < 100){</pre>
       converter=*freq2;
       converter= (converter/10) *10;
       if(*freq2 > converter) {
        converter=converter +10;
       *freq2=converter;
      }
     else{
       converter=*freq2;
       converter= (converter/100) *100;
       if (*freq2 > converter) {
         converter=converter +100;
         if (converter == 1000) {
           converter=1;
```

```
if (*g2=='H'){
            *g2='K';
       }
       else{
        *g2='M';
       }
        }
       }
       *freq2=converter;
        }
   }
}
void print_float(float freq)
{
int a;
int b;
   a=freq;
  print_num(10, a);
  send_char('.');
   b=(freq*10)-(a*10);
   print_num(10, b); // 1 casa decimal
  if(a < 10){
      send_char('0');
}
void print_long(unsigned char base, unsigned long val)
   unsigned char str[12];
   unsigned char i, digito;
   i = 0;
   do {
     digito = val % base;
     val /= base;
      str[i++] = bin2asc(digito);
   } while( val > 0 );
   while( i > 0 ) {
     while( TXIF == 0 );
     TXREG = str[--i];
   }
}
```

```
void config_spi(void)
   SSPSTAT=0x40; //escrita, transição ascendente
   SSPCON1=0x32; //SCKL= XTAL/64
//envio de n bytes
void send_byts(unsigned long val, char n)
{
   char lixo;
          do{
          n--;
          LATC0=0; //CS=0
          SSPBUF=(val >> (n*8));
          while(!BF);
          lixo=SSPBUF;
             LATC0=1; //CS=1;
          }while(n>0);
}
//escrita em um registo
void write_register(char regist, unsigned long val)
    switch(regist)
    case 0: send_byts(0x00,1);
            send_byts(val, 4); //cfr1
           break;
    case 1: send_byts(0x01,1);
           send_byts(val,3); //cfr2
           break;
    case 2: send_byts(0x02,1);
           send_byts(val,2); //ASF
           break;
    case 3: send_byts(0x03,1);
           send_byts(val,1); //ARR
           break;
    case 4: send_byts(0x04,1);
           send_byts(val, 4); //FTw0
           break;
    case 5: send_byts(0x05,1);
           send_byts(val, 2); //POWO
           break;
```

```
}
void dds_setup(void)
   reset=0; //high reset
   update=0;
   DelayMs(1);
   config_spi();
   DelayMs(1);
   reset=1; //high reset
   DelayMs(1);
   reset=0; //low reset
   DelayMs(1);
   write_register(CFR1, 0x00000000); //crf1
   write_register(CFR2,0x0000A4);
                                   //cfr2 pll 20
}
unsigned long freq_dds(float freq, char car)
unsigned long ft=0;
 switch (car)
   case 'H': freq= (freq * 4194304)/(732421.875);
            break;
   case 'K': freq= (freq * 4194304)/(732.421875);
            break;
   case 'M': freq= (freq * 4194304)/(0.732421875);
            break;
 ft=freq;
return ft;
}
void atenuadores (unsigned char atenuador, unsigned int val)
{
  PORTB=val;
  DelayMs(1);
  switch(atenuador)
```

```
case 0:
       LATA5=1;
       DelayMs(1);
       LATA5=0;
       break;
     case 1:
       LATD1=1;
       DelayMs(1);
       LATD1=0;
       break;
     case 2:
       LATD2=1;
       DelayMs(1);
       LATD2=0;
       break;
     case 3:
       LATD3=1;
       DelayMs(1);
       LATD3=0;
       break;
     case 4:
       LATD4=1;
       DelayMs(1);
       LATD4=0;
       break;
   DelayMs(1);
}
void estudo(unsigned long fx, char gx)
unsigned long val=0;
unsigned char i=1;
unsigned char j=0;
unsigned int atenuacao=0x3F;
unsigned int atenuacao2=0x3F;
float resultado=0;
unsigned int valrelay=0;
    Relay=1;
                         //sinal directo
    RelayIn=0;
                         //sinal opamp
    atenuadores(2, 0x3F); //atenuador canal 2 no maximo
val=freq_dds(fx, gx); //calcula o valor para programar o dds
escolhafiltros(fx, gx); //escolhe o filtro para a frequencia de trabalho
                            //canal 2 amplitude
    multiplex(2,1);
    write_register(FTW0, val); //programa o dds com a frequência desejada
    DelayMs(1);
                         //hight uptdate
    update=1;
        DelayMs(1);
                            //low uptdate
        update=0;
```

do{

```
val=ad8302();
       if(val<=10){
          if((atenuacao != 0) || (Relay != 0)) {
           if(atenuacao != 0) {
          atenuacao=atenuacao - 1;
             atenuadores(2, atenuacao);
        }
            else{
             atenuacao=0x3F;
             atenuadores(2, atenuacao);
                                   //Relay do canal passa para opamp
             Relay=0;
        }
          }
          else{
         if((atenuacao2 != 0) || (RelayIn != 0)) {
              if(atenuacao2 != 0) {
             atenuacao2= atenuacao2 - 1;
                atenuadores (0, atenuacao2);
              else{
             atenuacao2=0x3F;
            atenuadores (0, atenuacao2);
                RelayIn=0; //RelayIn passa para opamp
           }
         }
         else{
           send_char('-');
           print_num(10, 120); //Não foi possivel detectar envia -120dB
            print_num(10, 0); //Relays
            break;
         }
       }
           else{
         resultado=val;
         print_num(10, RelayIn+Relay); //Não foi possivel detectar envia -120dB
         if(resultado < 0) {
         send_char('-');
          else{
         send_char('0');
          print_float(resultado);
         }while(val<=10);</pre>
     send_char('E');
void calculo_calibracao(unsigned fx, char qx, unsigned long val)
```

```
float resultado=0;
unsigned int j=0;
  atenuadores(2, 0x00); //canal 2
  escolhafiltros(fx, gx); //escolhe o filtro para a frequencia de trabalho -
1Filtro
                                     //ftw0
  write_register(FTW0, val);
  DelayMs(1);
  update=1; //hight uptdate //aleterei
  DelayMs(1);
  update=0; //low uptdate //aleteri
  DelayMs(50);
val=0;
//lacombinacao
                         //sinal directo
  Relay=1;
                            //sinal directo
  RelayIn=1;
  j=0;
  do{
  DelayMs(50);
   val=ad8302();
                         //leitura da ADC
   resultado=resultado+val;
   j++;
   }while(j!=10);
  resultado=(resultado/76.74)-39.13; //Calcula o resultado
  if(resultado < 0) {      //Se o resultado for negativo imprime o caracter '-'</pre>
    send_char('-');
    resultado=-resultado;}
  print_float(resultado);
                            //Envia o resultado
  send_char('N');
//2ªcombinacao
  resultado=0;
                           //atenuador in no maximo
  atenuadores(0, 0x3F);
  DelayMs(50);
atenuadores(2, 0x00);
//sinal directo
                                 //atenuador canal 2 no minimo
  RelayIn=0;
                             //sinal opamp
  j=0;
  do{
   DelayMs(50);
   val=ad8302();
                        //leitura da ADC
   resultado=resultado+val;
   j++;
   }while(j!=10);
   resultado=(resultado/76.74)-41.89; //Calcula o resultado
  if(resultado < 0) {      //Se o resultado for negativo imprime o caracter '-'</pre>
    send_char('-');
```

```
resultado=-resultado;}
  send_char('N');
//3ªcombinacao
  resultado=0;
  atenuadores(0, 0x00);
                       //atenuador in no minimo
//atenuador canal 2 no maximo
  atenuadores(2, 0x3F);
                     //sinal opamp
  Relay=0;
                         //sinal directo
  RelayIn=1;
  j=0;
  do{
   DelayMs(50);
  val=ad8302(); //leitura da ADC
  resultado=resultado+val;
  j++;
  }while(j!=10);
  resultado=(resultado/76.74)-39.86; //Calcula o resultado
  if(resultado < 0) {</pre>
                          //Se o resultado for negativo imprime o caracter '-'
   send_char('-');
   resultado=-resultado;}
  print_float(resultado);
                         //Envia o resultado
  send_char('N');
//4combinacao
  resultado=0;
                       //atenuador in no maximo
  atenuadores(0, 0x3F);
  //atenuador canal 2 no maximo
  RelayIn=0;
                         //sinal opamp
  DelayMs(50);
  j=0;
  do{
   DelayMs(50);
   val=ad8302();
                      //leitura da ADC
   resultado=resultado+val;
   j++;
  }while(j!=10);
  resultado=(resultado/76.74)-42.62; //Calcula o resultado
  send_char('-');
   resultado=-resultado;}
                        //Envia o resultado
  print_float(resultado);
  send_char('N');
}
void calibracao(void)
{
```

```
resultado=-resultado; }
  send_char('N');
//3ªcombinacao
  resultado=0;
  RelayIn=1;
                       //sinal directo
  j=0;
  do{
   DelayMs(50);
  val=ad8302(); //leitura da ADC
  resultado=resultado+val;
  j++;
  }while(j!=10);
  resultado=(resultado/76.74)-39.86; //Calcula o resultado
  if(resultado < 0) {</pre>
                       //Se o resultado for negativo imprime o caracter '-'
   send_char('-');
   resultado=-resultado;}
  print_float(resultado);
                       //Envia o resultado
  send_char('N');
//4combinacao
  resultado=0;
  atenuadores(0, 0x3F); //atenuador in no maximo atenuadores(2, 0x3F); //atenuador canal 2 no maximo Relay=0; //sinal opamp
  RelayIn=0;
                       //sinal opamp
  DelayMs(50);
  j=0;
  do{
   DelayMs(50);
  val=ad8302();
                    //leitura da ADC
  resultado=resultado+val;
   j++;
  }while(j!=10);
  resultado=(resultado/76.74)-42.62; //Calcula o resultado
  send_char('-');
   resultado=-resultado;}
  send_char('N');
}
void calibracao(void)
{
```

```
unsigned int j=0;
unsigned long val=0;
float resultado=0.0;
//1°Filtro
    calculo_calibracao(5.5, 'K', 0x00007B08);
    calculo_calibracao(11,'K',0x0000F610);
   calculo_calibracao(16.5, 'K', 0x00017119);
    calculo_calibracao(22, 'K', 0x0001EC21);
   calculo_calibracao(33, 'K', 0x0005681E);
//2°Filtro
    calculo_calibracao(35, 'K', 0x00030EEF);
    calculo_calibracao(226.4, 'K', 0x0013C87B);
   calculo_calibracao(419.8, 'K', 0x0024AEC4);
    calculo_calibracao(646.2,'K',0x0038773F);
    calculo_calibracao(1,'M',0x0057619E);
//3°Filtro
    calculo_calibracao(3.3,'M',0x01205BC0);
    calculo_calibracao(5.5,'M',0x01E098EA);
   calculo_calibracao(7.8,'M',0x02A9930B);
   calculo_calibracao(10,'M',0x0369D036);
    calculo_calibracao(19.8, 'M', 0x0CAC0831);
//4°Filtro
    calculo_calibracao(28, 'M', 0x098EAD65);
    calculo_calibracao(46, 'M', 0x0FB38A94);
   calculo_calibracao(64, 'M', 0x15D867C3);
   calculo_calibracao(82, 'M', 0x1BFD44F3);
    calculo_calibracao(100, 'M', 0x22222222);
    atenuadores(0, 0x00);
                                    //atenuador in no minimo
    atenuadores(2, 0x00);
                                    //atenuador canal 2 no minimo
                               //canal 2 fase
        multiplex(2,0);
        escolhafiltros(1, 'M'); //escolhe o filtro para a frequencia de trabalho -
1Filtro
        write_register(FTW0,0x0057619E); //ftw0
        DelayMs(1);
        update=1; //hight uptdate
        DelayMs(1);
        update=0; //low uptdate
        DelayMs(50);
    j=0;
        do{
          DelayMs(50);
                               //leitura da ADC
          val=ad8302();
          resultado=resultado+val;
          j++;
        }while(j!=10);
    resultado=resultado/10;
    resultado=(resultado*3069)/1024;
```

```
resultado=(900.0 - resultado)/10;

if(resultado < 0.0) {    //Se o resultado for negativo imprime o caracter '-'
        send_char('-');
        resultado=-resultado;}

    if(resultado < 10.0) {
        send_char('0');
        send_char('0');}

if(resultado < 100.0) {
        send_char('0');}

print_num(10, resultado);
}</pre>
```

A.3 Driver.h

```
#include <pic18.h>
///////// USART
void usart_config(void);
void send_str(const char *pnt);
void send_char(char car);
char receive_char(void); void int_config(void);
void eco(void);
void put_str(unsigned char *str);
void print_num(unsigned char base, unsigned int val);
unsigned char bin2asc(unsigned char num);
unsigned int asc2dec(char aux);
unsigned int receive_int(void);
void val_string(char *str, unsigned int val);
unsigned int normalizacao(unsigned int val, unsigned int niveis, unsigned int vfe);
//////// DELAY
#ifndef XTAL_FREQ
#define XTAL_FREQ 20MHZ
                      /* Crystal frequency in MHz */
#endif
#define MHZ *1000L
                   /* number of kHz in a MHz */
#define KHZ *1
                /* number of kHz in a kHz */
#if XTAL_FREQ >= 12MHZ
#define DelayUs(x) { unsigned char _dcnt; \
          _dent = (x)*((XTAL_FREQ)/(12MHZ)); \
          while (--_dcnt != 0) \
             continue; }
#else
#define DelayUs(x) { unsigned char _dcnt; \
          _dent = (x)/((12MHZ)/(XTAL_FREQ))|1; 
          while (--_dcnt != 0) \
            continue; }
#endif
extern void DelayMs (unsigned char);
/* write a byte to the LCD in 4 bit mode */
```

A.3. DRIVER.H

```
extern void lcd_write(unsigned char);
/* Clear and home the LCD */
extern void lcd_clear(void);
/* write a string of characters to the LCD */
extern void lcd_puts(const char * s);
/* Go to the specified position */
extern void lcd_goto(unsigned char pos);
/* intialize the LCD - call before anything else */
extern void lcd_init(void);
extern void lcd_putch(char);
/* Set the cursor position */
#define lcd_cursor(x) lcd_write(((x)&0x7F)|0x80)
                         0x00 /* position of line1 */
0x40 /* position of line2 */
#define LINE1
#define
           LINE2
void lcd_started(void);
void lm335(void);
void config_ad8302(void);
unsigned long ad8302(void);
void fan(char);
void relay(char val);
void multiplex(char canal, char medida);
void pctopic(float *freq1, char *g1, float *freq2, char *g2, char *escala, int *amostras);
void limites(float *freq1, char *g1, float *freq2, char *g2);
void print_float(float);
void print_long(unsigned char base, unsigned long val);
void escolhafiltros(float freq, char gx);
void config_spi(void);
void send_byts(unsigned long val, char n);
void write_register(char regist, unsigned long val);
void dds_setup(void);
unsigned long freq_dds(float freq, char car);
void atenuadores (unsigned char atenuador, unsigned int val);
void estudo(unsigned long fx, char gx);
void calculo_calibracao(unsigned fx, char gx, unsigned long val);
void calibracao(void);
```

Apêndice B

Código Visual Basic

B.1 Form1.vb

```
Imports System.Windows.Forms
Imports System.Drawing
                                      'holds graphics, bitmap, brush, etc
Imports System.Drawing.Drawing2D 'holds smoothingmode, pentype enumerations
Imports System.Drawing.Imaging 'holds imageformat enumeration
Imports System.Math 'holds log function
Imports System.IO
                                     'holds file class for deleting existing files
Public Class Forml
    Inherits System.Windows.Forms.Form
    Public WithEvents rs232 As New MSCommLib.MSComm
Public str2 As String = ""
    Public freq() As Double
    Public ampl() As Double
    Public amp2() As Double
    Public amp3() As Double
    Public amp4() As Double
    Public fasel() As Double
    Public fase2() As Double
    Public fase3() As Double
Public fase4() As Double
    Public i As Integer = 0
    Dim conect As Integer = 0
    Public g As Char
Public g2 As Char
    Public escala As Char
    Public time As Integer = 0
    Public xData(4) As Double
Public yData(4) As Double
    Dim gfxscreen As Graphics
    Dim bmpscreen As Bitmap
    Dim timecounter = 0
    Dim tabela(0 To 3, 0 To 4, 0 To 3) As Decimal 'ntabela,linha,coluna Dim tabela2(0 To 3, 0 To 4, 0 To 3) As Decimal 'ntabela,linha,coluna
    Dim ntabela As Integer = 0
    Dim linha As Integer = 0
    Dim coluna As Integer = 0
    Dim accao As Integer = 0
    Dim tmp As String
    Dim count = 0
    Dim faseccalibracao As Integer
    Dim frequencias (0 To 20) As Double
    Private Sub Forml_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) ¥
    Handles MyBase.Load
         frequencias(0) = 5500
         frequencias(1) = 11000
         frequencias(2) = 16500
         frequencias(3) = 22000
         frequencias(4) = 33000
         frequencias(5) = 35000
         frequencias(6) = 226400
         frequencias(7) = 419800
         frequencias(8) = 646200
         frequencias(9) = 1000000
         frequencias(10) = 3300000
frequencias(11) = 5500000
         frequencias(12) = 7800000
         frequencias(13) = 10000000
         frequencias(14) = 19800000
         frequencias(15) = 28000000
         frequencias(16) = 46000000
         frequencias(17) = 64000000
         frequencias(18) = 82000000
         frequencias(19) = 100000000
         xData(0) = 1
```

```
xData(1) = 10
    xData(2) = 100
    xData(3) = 1000
xData(4) = 10000
    yData(0) = 0
yData(1) = -1
    yData(2) = -4
yData(3) = -40
    yData(4) = -40
    CheckBox1.Checked = False
    CheckBox2.Checked = False
CheckBox1.Enabled = False
    CheckBox2.Enabled = False
CheckBox3.Checked = True
    Label3.Enabled = False
    Labell.Visible = False
    Labell.Text = "Insira a frequência desejada: "
    TextBox1.Visible = False
    ListBoxl.Visible = False
    Label2.Visible = False
Label2.Text = ""
    TextBox2.Visible = False
    ListBox2.Visible = False
    Label4.Visible = False
    Label5.Visible = False
    CheckBox3.Checked = False
    CheckBox3.Visible = False
    CheckBox4.Checked = False
    CheckBox4.Visible = False
    TextBox3.Visible = False
    ListBox1.SelectedIndex = 0
    ListBox2.SelectedIndex = 0
    ComboBox1.Items.Clear()
    For Each sp As String In My.Computer.Ports.SerialPortNames
        ComboBox1.Items.Add(sp)
    Button1.Enabled = False
    Button7.Enabled = False
    Button6.Enabled = False
    Button4.Enabled = False
    Label7.Text = "Desconectado"
    Label7.ForeColor = Color.Red
    ParâmetrosToolStripMenuItem.Enabled = False SaveToolStripMenuItem.Enabled = True
    DataLogToolStripMenuItem.Enabled = False
    Timerl.Interval = 300
    ProgressBarl.Minimum = 0
End Sub
Sub escolha()
    Labell.Visible = True
    TextBoxl.Visible = True
    ListBoxl.Visible = True
```

```
Label4. Visible = True
     Label5.Visible = True
    CheckBox3.Checked = False
    CheckBox3.Visible = True
    CheckBox4.Checked = False
    CheckBox4.Visible = True
    TextBox3.Visible = True
End Sub
Private Sub TextBoxl TextChanged(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles TextBoxl.TextChanged
    If Len(TextBox1.Text) >= 1 Then
         If Asc(Strings.Right(TextBox1.Text, 1)) > 57 Then
             MessageBox.Show("Não introduza caracteres!!!", "Valor incorrecto")
TextBoxl.Text = "1"
         ElseIf Asc(Strings.Right(TextBox1.Text, 1)) < 48 Then
              If Asc(Strings.Right(TextBoxl.Text, 1)) <> 46 Then
    MessageBox.Show("Não introduza caracteres!!!", "Valor incorrecto")
    TextBoxl.Text = "1"
              End If
         End If
    End If
    If Val(TextBox1.Text) < 1 Then
    MessageBox.Show("Valor tem de ser positivo e menor que 166z!!!", "Valor</pre>
incorrecto")
         TextBox1.Text = ""
    End If
End Sub
Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles TextBox3.TextChanged
    If Len(TextBox3.Text) >= 1 Then
         If CheckBox3.Checked = True Then
              If Val(TextBox3.Text) > 1000 Then
    MessageBox.Show("Valor tem de ser menor que 1000!!!", "Valor
incorrecto")
                   TextBox3.Text = "2"
              ElseIf Val(TextBox3.Text) < 1 Then
                  MessageBox.Show("Valor tem de ser positivo!!!", "Valor incorrecto")
TextBox3.Text = "2"
              ElseIf Asc(Strings.Right(TextBox3.Text, 1)) > 57 Then
                   MessageBox.Show("Não introduza caracteres!!!",
                                                                           "Valor incorrecto")
                   TextBox3.Text = "2"
              ElseIf Asc(Strings.Right(TextBox3.Text, 1)) < 48 Then
    MessageBox.Show("Não introduza caracteres!!!", "Valor incorrecto")
    TextBox3.Text = "2"</pre>
              End If
         Text) > 1000 Then
                   MessageBox.Show("O numero máximo de amostras nao pode exceder 1000!!!" 🕊
, "Valor incorrecto")
                   TextBox3.Text = "2"
                   'End If
              ElseIf Val(TextBox3.Text) < 2 Then
                   MessageBox.Show("Valor tem de ser superior a 2!!!", "Valor incorrecto" ≰
                   TextBox3.Text = "2"
              ElseIf Asc(Strings.Right(TextBox3.Text, 1)) > 57 Then
    MessageBox.Show("Não introduza caracteres!!!", "Valor incorrecto")
                   TextBox3.Text = "2"
              ElseIf Asc(Strings.Right(TextBox3.Text, 1)) < 48 Then
    MessageBox.Show("Não introduza caracteres!!!", "Valor incorrecto")
    TextBox3.Text = "2"</pre>
```

```
End If
          End If
     End If
End Sub
Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles TextBox2.TextChanged
     If Len(TextBox2.Text) >= 1 Then
          If Val(TextBox2.Text) < 1 Then</pre>
              MessageBox.Show("Valor tem de ser positivo e menor que 166!!!", "Valor
incorrecto")
              TextBox2.Text = "1"
         ElseIf Asc(Strings.Right(TextBox2.Text, 1)) > 57 Then
    MessageBox.Show("Não introduza caracteres!!!", "Valor incorrecto")
    TextBox2.Text = "1"
          ElseIf Asc(Strings.Right(TextBox2.Text, 1)) < 48 Then</pre>
               If Asc(Strings.Right(TextBox2.Text, 1)) <> 46 Then
    MessageBox.Show("Não introduza caracteres!!!", "Valor incorrecto")
    TextBox2.Text = "1"
               End If
          End If
     End If
End Sub
Private Sub Buttonl Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 🛩
Handles Button1.Click
    Dim flag As Integer = 0
     If CheckBoxl.Checked = True Then
          If TextBox1.Text <> "" Then
               MessageBox.Show("PROGRAMING!!!", "DONE")
               grandeza()
               accao = 2
              rs232.SThreshold = 15
              rs232.Output = "F" & TextBox1.Text & g & "E"
              MessageBox. Show ("Parametros introuzidos incorrectos!!!", "ERRO")
          End If
     End If
     If CheckBox2.Checked = True Then
          If TextBox1.Text = "" Then
          MessageBox.Show("F1 errado!!!", "Dados incorrectos")
ElseIf TextBox2.Text = "" Then
         MessageBox.Show("F2 errado!!!", "Dados incorrectos")
ElseIf TextBox3.Text = "" Then
MessageBox.Show("Numero de pontos invalido!!!", "Dados incorrectos")
          ElseIf ListBoxl.SelectedIndex > ListBox2.SelectedIndex Then
              MessageBox.Show("F1 tem de ser menor que F2!!!", "Dados incorrectos")
          ElseIf ListBox1.SelectedIndex = ListBox2.SelectedIndex Then
              If Val(TextBox1.Text) >= Val(TextBox2.Text) Then
    MessageBox.Show("F1 tem de ser menor que F2!!!", "Dados incorrectos")
                   MessageBox.Show("PROGRAMING!!!", "DONE")
                   grandeza()
                   grandeza2()
                   escolha escala()
```

```
End If
        Else: MessageBox.Show("PROGRAMING!!!", "DONE")
             grandeza()
             grandeza2()
             escolha_escala()
        End If
    End If
    Dim dimensao(Val(TextBox2.Text) - 1 + 43) As Double
    ProgressBarl.Maximum = Val(TextBox2.Text)
    freq = dimensao
    ampl = dimensao
    amp2 = dimensao
    amp3 = dimensao
    amp4 = dimensao
    fasel = dimensao
    fase2 = dimensao
    fase3 = dimensao
    fase4 = dimensao
End Sub
Private Sub ListBoxl SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ListBox1.SelectedIndexChanged
    If ListBox1.SelectedIndex = 2 Then
    If Val(TextBox1.Text) > 166 Then
        MessageBox.Show("Valor tem de ser positivo e menor que 166!!!", "Valor
incorrecto")
             TextBox1.Text = ""
        End If
    Else
        If Val(TextBox1.Text) > 999 Then
    MessageBox.Show("Valor tem de ser positivo e menor que 166!!!", "Valor
End If
    End If
    grandeza()
End Sub
Private Sub ListBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ListBox2.SelectedIndexChanged
    If ListBox2.SelectedIndex = 2 Then
        If Val(TextBox2.Text) > 166 Then
    MessageBox.Show("Valor tem de ser positivo e menor que 166!!!", "Valor
                                                                                            ~
incorrecto")
             TextBox2.Text = ""
        End If
    Else
        If Val(TextBox2.Text) > 999 Then
            MessageBox. Show ("Valor tem de ser positivo e menor que 166!!!", "Valor
End If
    End If
End Sub
Private Sub CheckBoxl_CheckedChanged(ByVal sender As System.Object, ByVal e As System.⊌
EventArgs) Handles CheckBoxl.CheckedChanged
```

```
If CheckBoxl.Checked = True Then
        CheckBox2.Checked = False
        escolha()
        Labell.Text = "Insira a frequência desejada: "
        Label2.Visible = False
        TextBox2.Visible = False
        ListBox2.Visible = False
        Label4.Visible = False
Label5.Visible = False
        CheckBox3.Checked = False
        CheckBox3.Visible = False
        CheckBox4.Checked = False
CheckBox4.Visible = False
        TextBox3.Visible = False
        Button1.Enabled = True
    End If
Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.⊭
EventArgs) Handles CheckBox2.CheckedChanged
    If CheckBox2.Checked = True Then
    CheckBox1.Checked = False
        escolha()
        Labell.Visible = True
Labell.Text = "Insira a l frequência desejada: "
        TextBoxl.Visible = True
        ListBoxl.Visible = True
        Label2.Visible = True
Label2.Text = "Insira a 2 frequência desejada: "
        TextBox2.Visible = True
        ListBox2.Visible = True
        Label5.Text = "Numero de pontos por decada: "
        CheckBox3.Checked = True
        Button1.Enabled = True
        accao = 3
    End If
End Sub
Private Sub CheckBox3 CheckedChanged(ByVal sender As System.Object, ByVal e As System.⊌
EventArgs) Handles CheckBox3.CheckedChanged
    If CheckBox3.Checked = True Then
        CheckBox4.Checked = False
    End If
End Sub
Private Sub CheckBox4_CheckedChanged(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles CheckBox4.CheckedChanged
    If CheckBox4.Checked = True Then
        CheckBox3.Checked = False
    End If
End Sub
Private Sub Button6 Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 🛩
Handles Button6.Click
    rs232.Settings = "57600,N,8,1" 'Startup of rs232
    rs232.InputLen = 0
    rs232.RThreshold = 1
    rs232.PortOpen = True
    'Sempre que um caracter chega é gerado um invento
    conect = 0
    Button6.Enabled = False
    rs232.Output = "E"
```

```
accao = 0
    time = 1
    Timerl.Start()
End Sub
Private Sub Button7 Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 🕊
Handles Button7.Click
    Button1.Enabled = False
    Button6.Enabled = True
    Button7.Enabled = False
str2 = ""
    ComboBox1.Enabled = True
    rs232.PortOpen = False
Label7.Text = "Desconectado"
Label7.ForeColor = Color.Red
End Sub
Private Sub ComboBoxl_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As 🛩
System.EventArgs) Handles ComboBoxl.SelectedIndexChanged
    Button6.Enabled = True
    str2 = ComboBox1.SelectedItem.ToString
    opcao()
End Sub
Sub opcao()
    Dim porta As Char
    Dim i As Integer
    If str2.Length < 5 Then
        str2 = Microsoft.VisualBasic.Right(str2, 1)
         TextBoxl.Text = Asc(str2) - 48
        rs232.CommPort = Asc(str2) - 48
        porta = Microsoft.VisualBasic.Mid(str2, 4, 1)
         i = (Asc(porta) - 48) * 10
porta = Microsoft.VisualBasic.Right(str2, 1)
i = i + (Asc(porta) - 48)
        rs232.CommPort = i
    End If
End Sub
Private Sub EXITToolStripMenuItem Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles EXITToolStripMenuItem.Click
    Dim sair As DialogResult
    sair = MessageBox.Show("Tem a certeza que pertende sair?", "EXIT",
MessageBoxButtons.YesNoCancel)
    If sair = Windows.Forms.DialogResult.Yes Then
         'Se se pretender sair do programa e a porta COM estiver aberta, esta é fechada If rs232.PortOpen = True Then
             rs232.PortOpen = False
         End If
        Application.Exit()
    End If
End Sub
Private Sub rs232 OnComm() Handles rs232.OnComm
    Dim a As String
    a = rs232.Input
    Select Case accao
Case 0 'Coneccao pic <-> pc
             If a = "A" Then 'Se recebeu A (Aknoledge) o pic respondeu
                  Timerl.Stop()
                  Label7.Text = "Conectado"
                  Label7.ForeColor = Color.Green
```

```
conect = 1
                Button7.Enabled = True
                Button4.Enabled = True
                Button6.Enabled = False
                ComboBox1.Enabled = False
                rs232.RThreshold = 6
            End If
        Case 1 'Calibração amplitude
            If Strings.Right(a, 1) = "N" Then
                tabela(ntabela, linha, coluna) = -22.6 - CDec(Val(Strings.Left(a, 5)))
                Select Case count
                    Case 0
                        graficocalibracao.TextBox3.Text ω= Str(tabela(ntabela, linha, κ
coluna)) & vbCrLf
                    Case 1
                        graficocalibracao.TextBox4.Text ω= Str(tabela(ntabela, linha, κ
coluna)) & vbCrLf
                    Case 2
                        graficocalibracao.TextBox5.Text &= Str(tabela(ntabela, linha, 🗹
coluna)) & vbCrLf
                    Case 3
                       graficocalibracao.TextBox6.Text &= Str(tabela(ntabela, linha, ¥
coluna)) & vbCrLf
                End Select
                count = count + 1
                If count = 4 Then
                    count = 0
                End If
                'graficocalibracao.TextBox3.Text &= Str(tabela(ntabela, linha,
coluna)) & vbCrLf
                coluna = coluna + 1
                If coluna = 4 Then
                    coluna = 0
                    linha = linha + 1
                End If
                If linha = 5 Then
                    linha = 0
                    ntabela = ntabela + 1
                If ntabela = 4 Then
                    rs232.RThreshold = 3
                    CheckBox1.Enabled = True
                    CheckBox2.Enabled = True
                    Label3.Enabled = True
                    Button4.Enabled = False
                    graficocalibracao.Show()
                    accao = 4
                End If
            End If
        Case 2 'Observar frequencia
            tmp s = a
            If Len(tmp) = 18 Then
                dados.TextBoxl.Text &= "Frequência: " & Strings.Mid(tmp, 2, 6)

If Strings.Mid(tmp, 7, 1) = "H" Then

dados.TextBoxl.Text &= "z" & Chr(9)
                Else
                    dados.TextBox1.Text &= "Hz" & Chr(9)
                End If
               faseccalibracao) & " Graus" & Chr(9)
                dados.TextBoxl.Text &= "Amplitude: " & Strings.Mid(tmp, 11, 5) & " dB" ¥
```

```
& Chr(9)
                   dados.TextBox1.Text &= "Relay: " & Strings.Mid(tmp, 16, 1)
                   dados.Show()
              End If
         Case 3 'Efectuar Sweep
              freq(i) = Strings.Mid(a, 2, 4)
              ampl(i) = Strings.Mid(a, 6, 4)
              'amp2(i) = Strings.Mid(a, 13, 4)
'fase2(i) = Strings.Mid(a, 17, 3)
'amp3(i) = Strings.Mid(a, 20, 4)
'fase3(i) = Strings.Mid(a, 24, 3)
'amp4(i) = Strings.Mid(a, 27, 4)
'fase4(i) = Strings.Right(a, 3)
              ProgressBarl.Value = i
              i = i + 1
              If i = Val(TextBox3.Text) Then
    MessageBox.Show("OPERACÃO COMPLETA", "FIM", MessageBoxButtons.OK)
         Case 4 'Calibração fase
              faseccalibracao = a + 4
     End Select
End Sub
Private Sub ParametroSllToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e≰
 As System.EventArgs)
     Graficos.Show()
    Graficos.WindowState = FormWindowState.Maximized
End Sub
Private Sub DataLogToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As & System.EventArgs) Handles DataLogToolStripMenuItem.Click
     dados.Show()
End Sub
Private Sub Timerl_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timerl.Tick
     Timer1.Stop()
     If time = 3 Then
         time = 1
         rs232.PortOpen = False
         MessageBox.Show("Impossível Conectar", "ERRO", MessageBoxButtons.OK)
         Button6.Enabled = True
     Else
          time = time + 1
          rs232.Output = "E"
         Timerl.Start()
     End If
End Sub
Private Sub ResultadosToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e 

▲ As System.EventArgs) Handles ResultadosToolStripMenuItem.Click
     If i = Val(TextBox3.Text) Then
         ParâmetrosToolStripMenuItem.Enabled = True
          SaveToolStripMenuItem.Enabled = True
         DataLogToolStripMenuItem.Enabled = True
```

```
End If
End Sub
Private Sub ProgressBarl_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles ProgressBarl.Click
Sub grandeza()
    If ListBox1.SelectedIndex = 0 Then g = "H" End If
     If ListBox1.SelectedIndex = 1 Then
         g = "K"
     If ListBox1.SelectedIndex = 2 Then
     g = "M"
End If
End Sub
Sub grandeza2()
     If ListBox2.SelectedIndex = 0 Then
    g2 = "H"
End If
     If ListBox2.SelectedIndex = 1 Then
     g2 = "K"
End If
     If ListBox2.SelectedIndex = 2 Then
     g2 = "M"
End If
End Sub
Sub escolha_escala()
     If CheckBox3.Checked Then
          escala = "I'
         escala = "O"
     End If
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
End Sub
Private Sub SaveToolStripMenuItem_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SaveToolStripMenuItem.Click
SaveFileDialog1.Title = "Guardar dados"
SaveFileDialog1.FileName = "resultados"
SaveFileDialog1.Filter = "Text file (*.txt) |*.txt"
If SaveFileDialog1.ShowDialog() = DialogResult.OK Then
          Dim SW As New IO.StreamWriter(SaveFileDialog1.FileName, True) dados.TextBox1.Text = DateString
          dados.TextBoxl.Text \&= vbCrLf dados.TextBoxl.Text \&= "2ramo"
          SW.Write(dados.TextBox1.Text)
          SW.Close()
          MessageBox.Show("Dados guardados com sucesso!!!", "DONE")
          Timer3.Enabled = True
          Timer3.Start()
     End If
End Sub
```

```
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer2.Tick
    Timer2.Stop()
    bmpscreen = New Bitmap (Graficos. Size. Width, Graficos. Size. Height, Pixel Format.
Format32bppArgb)
    gfxscreen = Graphics.FromImage(bmpscreen)
    gfxscreen.CopyFromScreen(Graficos.Location.X, Graficos.Location.Y, 0, 0, Graficos.⊌
Size, CopyPixelOperation.SourceCopy)
    bmpscreen.Save(SaveFileDialogl.FileName, ImageFormat.Png)
   MessageBox.Show("Gracficos guardados com sucesso!!!", "DONE")
   Graficos.Close()
    If (SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK) Then
        Graficos2.Show()
        Graficos2.WindowState = FormWindowState.Maximized
        Timer4.Enabled = True
        Timer4.Start()
    End If
End Sub
Private Sub ParâmetrosToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e 🕏 💆
As System. EventArgs) Handles Parâmetros Tool Strip Menu Item. Click
    Graficos.Show()
End Sub
Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer3.Tick
    Timer3.Stop()
    SaveFileDialog1.Title = "Guardar Graficos"
    SaveFileDialog1.FileName = "graficos"
    SaveFileDialog1.Filter = "Formato de Imagem JPEG|*.jpeg|Formato de Imagem GIF|*.
gif|Formato de Imagem PNG|*.png
    If (SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK) Then
        Graficos.Show()
        Graficos.WindowState = FormWindowState.Maximized
        Timer2.Enabled = True
        Timer2.Start()
    End If
End Sub
Private Sub Timer4_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer4.Tick
   Timer4.Stop()
   bmpscreen = New Bitmap(Graficos2.Size.Width, Graficos2.Size.Height, PixelFormat.
Format32bppArgb)
    gfxscreen = Graphics.FromImage(bmpscreen)
    gfxscreen.CopyFromScreen(Graficos2.Location.X, Graficos2.Location.Y, 0, 0,
Graficos2.Size, CopyPixelOperation.SourceCopy)
    bmpscreen.Save(SaveFileDialog1.FileName, ImageFormat.Png)
   MessageBox.Show("Gracficos guardados com sucesso!!!", "DONE")
   Graficos2.Close()
Private Sub Button4 Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 🕊
Handles Button4.Click
   Button4.Enabled = False
    accao = 1
    rs232.Output = "C"
End Sub
Sub compensacao(ByVal freq, ByVal grandeza, ByRef parametro, ByVal relay)
    Dim valor As Double
    Select Case g
        Case "K"
           freq = freq * 1000
        Case "M"
           freq = freq * 1000000
    End Select
```

```
For xxx = 0 To 19
            If (valor > freq) Then
                If xxx = 0 Then
                    parametro = parametro + tabela(0, 0, relay)
                   parametro = parametro + tabela((xxx - 1) / 4, (xxx - 1) / 5, relay)
                End If
            Else
            End If
        Next xxx
    End Sub
    Private Sub VerGraficosToolStripMenuItem Click(ByVal sender As System.Object, ByVal e 🛩
    As System.EventArgs) Handles VerGraficosToolStripMenuItem.Click
        Dim gfxscreen As Graphics
        Dim bmpscreen As Bitmap
        If (SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK) Then
            bmpscreen = New Bitmap (Me.Size.Width, Me.Size.Height, PixelFormat.
    Format32bppArgb)
            gfxscreen = Graphics.FromImage(bmpscreen)
            gfxscreen.CopyFromScreen(Me.Location.X, Me.Location.Y, 0, 0, Me.Size,
    CopyPixelOperation.SourceCopy)
           bmpscreen.Save(SaveFileDialogl.FileName, ImageFormat.Png)
        Graficos.Show()
        Graficos2.Show()
    End Sub
End Class
```

B.2 Graficocalibração.vb

```
Private Sub graficocalibracao Load (ByVal sender As System.Object, ByVal e As System. 
EventArgs) Handles MyBase.Load

TextBox2.Text = "5.5KHz" & vbCrLf

TextBox2.Text s= "11KHz" & vbCrLf

TextBox2.Text s= "16.5KHz" & vbCrLf

TextBox2.Text s= "22KHz" & vbCrLf

TextBox2.Text s= "33KHz" & vbCrLf

TextBox2.Text s= "33KHz" & vbCrLf

TextBox2.Text s= "35KHz" & vbCrLf

TextBox2.Text s= "35KHz" & vbCrLf

TextBox2.Text s= "419.8KHz" & vbCrLf

TextBox2.Text s= "419.8KHz" & vbCrLf

TextBox2.Text s= "1MHz" & vbCrLf

TextBox2.Text s= "3.3MHz" & vbCrLf

TextBox2.Text s= "3.3MHz" & vbCrLf

TextBox2.Text s= "7.8MHz" & vbCrLf

TextBox2.Text s= "7.8MHz" & vbCrLf

TextBox2.Text s= "10MHz" & vbCrLf

TextBox2.Text s= "10MHz" & vbCrLf

TextBox2.Text s= "46MHz" & vbCrLf

TextBox2.Text s= "46MHz" & vbCrLf

TextBox2.Text s= "64MHz" & vbCrLf

TextBox2.Text s= "82MHz" & vbCrLf
```

B.3 Grafico.vb

End Property

```
Imports System.Windows.Forms
Imports Dundas.Charting.WinControl
Imports System.Drawing
                                     'holds graphics, bitmap, brush, etc
Imports System.Drawing.Drawing2D 'holds smoothingmode, pentype enumerations
Imports System.Drawing.Imaging 'holds imageformat enumeration
Imports System.Math 'holds log function
Imports System.IO
                                    'holds file class for deleting existing files
Public Class Graficos
    Dim i = 0
#Region " Private Declarations "
    Private x0 As Integer
    Private xl As Integer
    Private y0 As Integer
    Private yl As Integer
    Private w0 As Integer
    Private wl As Integer
    Private h0 As Integer
    Private hl As Integer
    Private x As Integer
    Private y As Integer
    Private n As Integer
    Private d As Integer
    Private dl As Integer
    Private s As String
#End Region
#Region " Public Properties "
    Private _colorDraw As Color = Color.DarkBlue
Public Property ColorDraw() As Color
         Get
             Return _colorDraw
         End Get
Set(ByVal value As Color)
        _colorDraw = value
End Set
    End Property
    Private colorGrid As Color = Color.LightGray
    Public Property ColorGrid() As Color
             Return _colorGrid
         End Get
         Set (ByVal value As Color)
         _colorGrid = value
End Set
    End Property
              _colorBg As Color = Color.White
    Private
    Public Property ColorBg() As Color
         Get
             Return _colorBg
         End Get
         Set(ByVal value As Color)
              colorBg = value
         End Set
    End Property
    Private _colorAxis As Color = Color.Black Public Property ColorAxis() As Color
             Return _colorAxis
         End Get
         Set (ByVal value As Color)
        __colorAxis = value
End Set
```

B.3. GRAFICO.VB

```
Private _fontAxis As Font = New Font("Arial", 8)
Public Property FontAxis() As Font
    Get
         Return _fontAxis
    End Get
    Set(ByVal value As Font)
    _fontAxis = value
End Set
End Property
Private penWidth As Integer = 2
Public Property PenWidth() As Integer
    Get
         Return _penWidth
    End Get
Set(ByVal value As Integer)
    _penWidth = value
End Set
End Property
          borderTop As Integer = 30
Public Property BorderTop() As Integer
         Return _borderTop
    End Get
Set(ByVal value As Integer)
          borderTop = value
    End Set
End Property
Private borderLeft As Integer = 50
Public Property BorderLeft() As Integer
        Return borderLeft
    End Get
    Set(ByVal value As Integer)
    _borderLeft = value
End Set
End Property
Private _borderBottom As Integer = 30
Public Property BorderBottom() As Integer
        Return borderBottom
    End Get
    Set(ByVal value As Integer)
          borderBottom = value
    End Set
End Property
Private _borderRight As Integer = 30
Public Property BorderRight() As Integer
    Get
         Return _borderRight
    End Get
    Set(ByVal value As Integer)
    _borderRight = value
End Set
End Property
Private _xRangeStart As Double = 0.1
Public Property XRangeStart() As Double
    Get
         Return _xRangeStart
    End Get
    Set(ByVal value As Double)
    \underline{\phantom{a}}xRangeStart = value
End Set
End Property
Private _xRangeEnd As Double = 1000000000 Public Property XRangeEnd() As Double
    Get
```

```
Return _xRangeEnd
        End Get
        Set(ByVal value As Double)
        _xRangeEnd = value
End Set
    End Property
              yRangeStart As Double = 0
    Public Property YRangeStart() As Double
             Return _yRangeStart
        End Get
        Set(ByVal value As Double)
        yRangeStart = value
End Set
    End Property
    Private _yRangeEnd As Double
Public Property YRangeEnd() As Double
            Return _yRangeEnd
         End Get
        Set(ByVal value As Double)
        _yRangeEnd = value
End Set
    End Property
    Private xGrid As Integer = 100
    Public Property XGrid() As Integer
        Get
             Return _xGrid
        End Get
Set(ByVal value As Integer)
              xGrid = value
        End Set
    End Property
    Private _yGrid As Integer
Public Property YGrid() As Integer
        Get
            Return _yGrid
        End Get
        Set(ByVal value As Integer)
        _yGrid = value
End Set
    End Property
    Private _xLogBase As Integer = 10
Public Property XLogBase() As Integer
        Get
             Return _xLogBase
         End Get
        Set(ByVal value As Integer)
        _xLogBase = value
End Set
    End Property
    Private _yLogBase As Integer = 0
Public Property YLogBase() As Integer
        Get
            Return _yLogBase
        End Get
Set(ByVal value As Integer)
              yLogBase = value
        End Set
    End Property
#End Region
    Public Sub Render(ByVal xData() As Double, ByVal yData() As Double, ≰
    ByRef picturebox As PictureBox, ByVal ampl As Integer)
         Dim outputBitmap As New Bitmap (600, 300)
         Dim g As Graphics = Graphics.FromImage(outputBitmap)
        g.SmoothingMode = SmoothingMode.AntiAlias
```

B.3. GRAFICO.VB

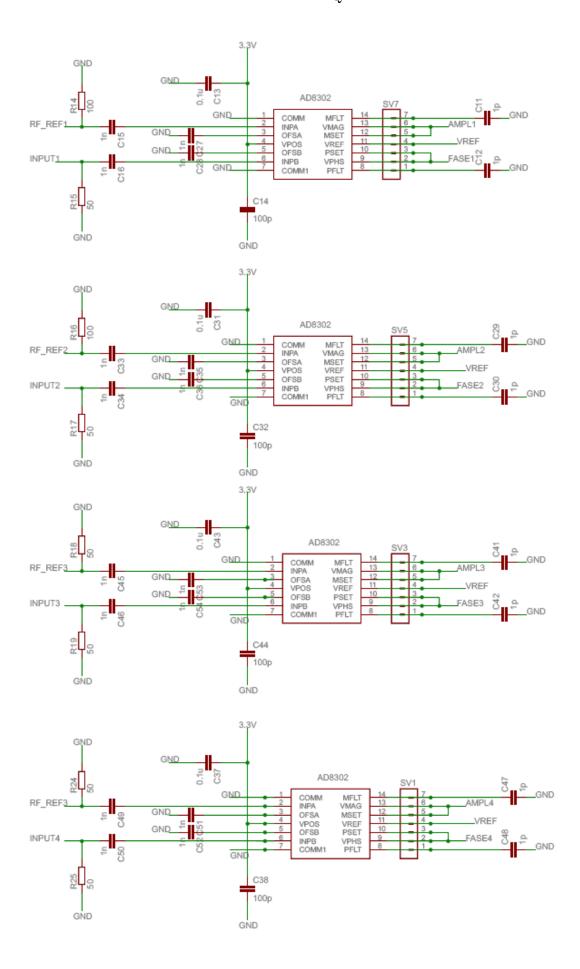
```
Dim clientRectangle As New RectangleF(0, 0, 600, 300)
    x0 = clientRectangle.Left + BorderLeft
    y0 = clientRectangle.Top + BorderTop
    w0 = clientRectangle.Width - BorderLeft - BorderRight
    h0 = clientRectangle.Height - BorderTop - BorderBottom
x1 = clientRectangle.Right - BorderRight
y1 = clientRectangle.Bottom - BorderBottom
    g.FillRectangle(New SolidBrush(ColorBg), clientRectangle)
    Me.DrawVerticalLines(g)
    w1 = d * n
    Me.DrawHorizontalLines(g, ampl)
    Dim penAxis As New Pen(ColorAxis, 1)
    h1 = d * n
    g.DrawRectangle(penAxis, x0, y0, w0, h0) ' draw axis
    h0 = h1 'must correct internal width & height since equidistant w0 = w1 'gridlines may not fit in axis rectangle w/o rounding errors
    Me.DrawData(g, xData, yData)
    picturebox. Image = outputBitmap
End Sub
Public Sub DrawVerticalLines(ByVal g As Graphics)
    Dim penGrid As New Pen(ColorGrid, 1)
Dim brushAxis As New SolidBrush(ColorAxis)
    n = Convert.ToInt32(Math.Log(XRangeEnd, XLogBase) - Math.Log(XRangeStart,
XLogBase)) 'get the x width converted to log10
    If n = 0 Then n = 1
    d = w0 / n
    For i As Integer = 0 To n
        x = x0 + i * d
If i < n Then
              For j As Integer = 1 To XLogBase - 1
                  dl = Convert.ToInt32(Math.Log(j, XLogBase) * d)
                  g.DrawLine(penGrid, x + dl, y0, x + dl, y1)
             Next
         End If
         s = Me.LargeFormat (Convert.ToString (Math.Pow (XLogBase, Math.Log (XRangeStart, #
XLogBase) + i)))
         Dim sf As SizeF = g.MeasureString(s, FontAxis)
         g.DrawString(s, FontAxis, brushAxis, x - sf.Width / 2, y1 + sf.Height / 2)
End Sub
Public Function LargeFormat(ByVal value As String) As String
    Dim result As String
    Select Case value
         Case "1000"
             result = "1K"
         Case "10000"
             result = "10K"
         Case "100000"
             result = "100K"
         Case "1000000"
             result = "1M"
         Case "10000000"
             result = "10M"
         Case "1000000000"
             result = "100M"
         Case Else
             result = value
    End Select
    Return result
End Function
Public Sub DrawHorizontalLines(ByVal g As Graphics, ByVal ampl As Integer)
    Dim s As String
    Dim penGrid As New Pen(ColorGrid, 1)
    Dim brushAxis As New SolidBrush (ColorAxis)
    ' n = Convert.ToInt32((YRangeEnd - YRangeStart) / YGrid)
'n = Convert.ToInt32(7)
    n = 10
    If n = 0 Then n = 1
```

```
d = h0 / n
    For i As Integer = 0 To n
       y = yl - i * d
        g.DrawLine(penGrid, x0, y, x1, y)
        If ampl = 1 Then
           s = Convert.ToString(-210 + (i * 30))
        Else
           s = Convert.ToString(-90 + (i * 10))
        End If
        Dim sf As SizeF = g.MeasureString(s, FontAxis)
       g.DrawString(s, FontAxis, brushAxis, x0 - sf.Width - sf.Height / 4, y - sf.
Height / 2)
   Next
End Sub
Public Sub DrawData(ByVal g As Graphics, ByVal xdata() As Double, ByVal ydata() As
Double)
    Dim penDraw As New Pen(ColorDraw, PenWidth)
    Dim pts(xdata.Length - 1) As Point
Dim lastValidPt As New Point(x0, y1)
    For i As Integer = 0 To pts.Length - 1 'convert points to fit inside graphing area
            pts(i).X = Convert.ToInt32(x0 + (Math.Log(xdata(i), XLogBase) - Math.Log w
(XRangeStart, XLogBase)) / (Math.Log(XRangeEnd, XLogBase) - Math.Log(XRangeStart,
XLogBase)) * w0)
            pts(i).Y = Convert.ToInt32(yl - (ydata(i) + YRangeStart + 60) / (10 + 60) #
           lastValidPt = pts(i)
        Catch ex As Exception
   pts(i) = lastValidPt 'redraw last valid point on error
End Try
    Next
    For i As Integer = 0 To pts.Length - 1 'now draw the points
        If i > 0 Then g.DrawLine(penDraw, pts(i - 1), pts(i))
   Next
End Sub
Private Sub S11_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
   Render(Forml.xData, Forml.yData, Graficos2.PictureBox1, 0)
End Sub
```

End Class

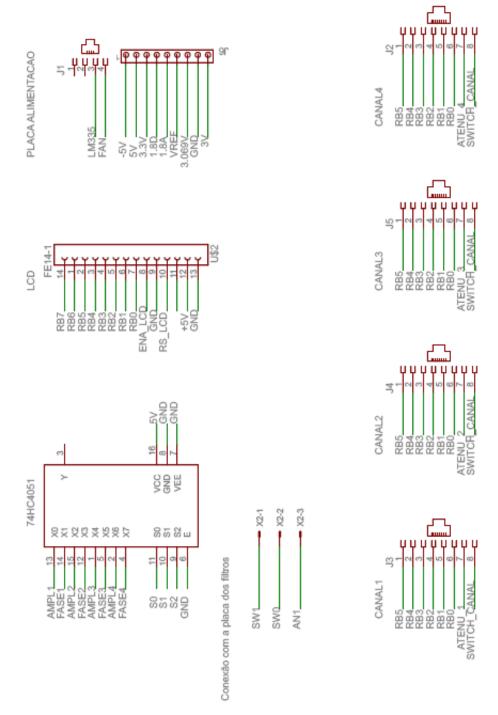
Apêndice C

Esquemático Motherboard



a3

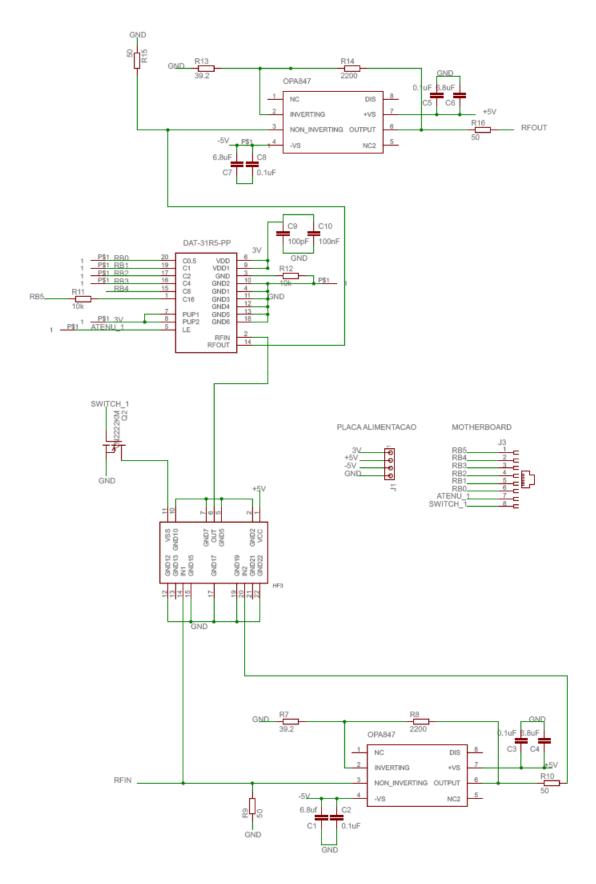
a3 costas



Apêndice D

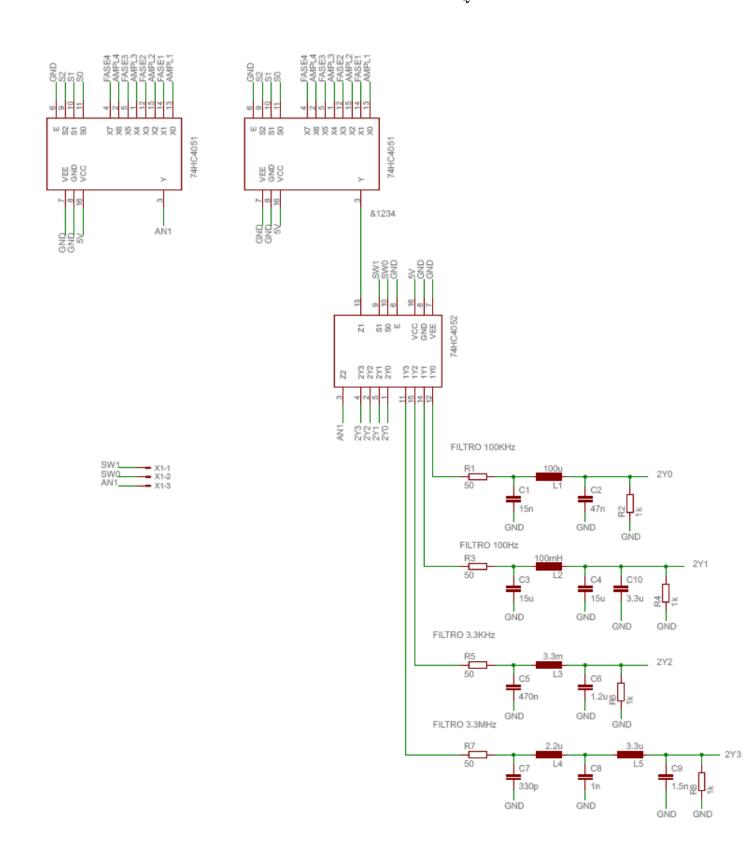
Esquemático Placa pré amplificador de entrada

$136AP \hat{E}NDICE\,D.\,\,ESQUEM \\ \acute{A}TICO\,PLACA\,PR \\ \acute{E}\,AMPLIFICADOR\,DE\,ENTRADA$



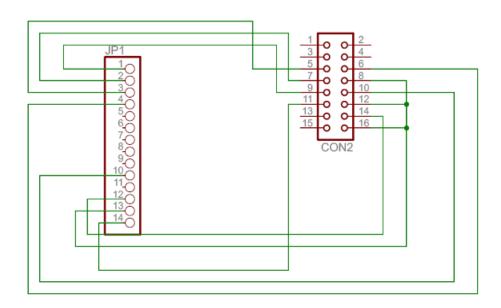
Apêndice E

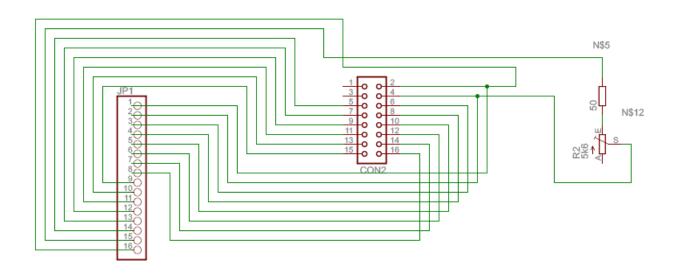
Esquemático Filtros



Apêndice F

Esquemático LCD





Apêndice G

Esquemático Placa de Alimentação

a3 costas

Apêndice H

Hardware total

