



**Universidade do Minho**  
Escola de Engenharia

Rafael Augusto Cunha Costinha Névoa

***Toolbox* para Plataforma de Teste de  
MEMS**

Dissertação de Mestrado  
Mestrado Integrado em Engenharia Eletrónica Industrial  
e Computadores

Trabalho efetuado sobre a orientação do  
**Professor Doutor Jorge Cabral**

Outubro de 2017

## **DECLARAÇÃO**

**Nome:** Rafael Augusto Cunha Costinha Névoa

**Endereço eletrónico:** rafael.accn@gmail.com   **Telefone:** 917973710

**Cartão do Cidadão:** 14376553

**Título da dissertação:** Toolbox para Plataforma de Teste de MEMS

**Orientador:** Professor Doutor Jorge Miguel Nunes Santos Cabral

**Ano de conclusão:** 2017

Mestrado em Engenharia Eletrónica Industrial e Computadores

É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA DISSERTAÇÃO, APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, \_\_\_\_/\_\_\_\_/\_\_\_\_\_

Assinatura:

# Agradecimentos

Em primeiro lugar queria agradecer aos meus pais, Augusto e Fátima e à minha irmã Lara, por tudo o que têm feito por mim ao longo da minha vida, em particular no meu percurso académico.

Ao Professor Doutor Luís Rocha e ao Professor Doutor Jorge Cabral, pelo apoio e confiança depositados em mim para a realização desta dissertação.

Aos meus colegas de curso e laboratório, que tornaram o desenvolvimento deste projeto possível, por todo o tempo e paciência dispensados, ajudando-me e apoiando-me sempre.

Aos meus colegas de curso e amigos de sempre, por todos os bons momentos passados ao longo deste percurso, pelas horas passadas no “covil”, em trabalho e em lazer, um obrigado especial.

Por fim, mas não menos importante, à minha namorada, Beatriz, pela paciência, pelo apoio e por me fazer ver sempre o lado positivo das coisas, mesmo nos momentos mais frustrantes.

O meu obrigado a todos!





## Resumo

Com o crescimento do mercado de sistemas microelectromecânicos (MEMS), praticamente todos os grandes fabricantes de semicondutores têm as suas próprias linhas de produção de MEMS e enfrentam constantes aumentos de volume de produção para acompanhar a demanda destes [1]. Todos estes sistemas têm de ser testados após a produção, para caracterização das estruturas, a fim de obter as especificações da mesma, para calibração em processo fabril ou para verificação e validação do *design*. A oferta deste tipo de sistemas de teste é reduzida. Devido a este défice, e com o intuito de solucionar este problema, foi iniciado, na Universidade do Minho, o processo de construção de uma plataforma de teste de MEMS [2]. Atualmente, o *hardware* desta já foi desenvolvido, sendo agora necessário criar um conjunto de ferramentas que permitam ao utilizador interagir com a plataforma através de uma interface simples e intuitiva, garantindo uma comunicação rápida e fiável com o sistema de testes e expandindo as funcionalidades da mesma, através de blocos de *software* que implementem rotinas específicas de teste de MEMS.

Com esta dissertação, pretende-se criar o *software* que solucione o problema anteriormente mencionado, passando a solução escolhida por dividir a *Toolbox* em três componentes principais: o *driver* de comunicação com a plataforma de testes utilizado para definir e converter os comandos do instrumento, a interface gráfica a desenvolver em MATLAB e as rotinas específicas de teste de MEMS.

Como resultado do trabalho desenvolvido foi possível criar um sistema que permite ao utilizador interagir com a plataforma de teste de MEMS [2] de uma forma simples e intuitiva, com a fiabilidade de um *driver* de comunicação e expandindo as funcionalidades deste sistema, em termos de mecanismos de aquisição e teste.

Palavras-Chave: *Toolbox*, Teste de MEMS, GUI, *Driver*, MATLAB



# Abstract

With the growth of the microelectromechanical systems (MEMS) market, nearly all of the main manufacturers of semiconductors have their own MEMS production lines, and have to deal with constant increases in production volumes to keep up with market demand [1]. All these systems must be tested post production, for structure characterization, in order to obtain the key parameters used, for calibration during the manufacturing process, or for design verification and validation. The offer for these types of structure characterization systems is reduced.

Due to this deficit – and to solve this problem – the development process of a MEMS test platform [2] has started in the University of Minho. Currently, its hardware has already been developed, so it was necessary to create a toolbox that allows the user to easily use the platform through a simple and intuitive interface, thus allowing a fast and reliable communication with the test system and expanding its functionalities, through software blocks that implement specific MEMS test routines.

This dissertation aims to create the software to solve the previously mentioned problem. With this objective in mind, the chosen solution requires that the toolbox is divided in three main components: the driver to communicate with the test platform, define and convert the instrument commands, the graphic interface designed using MATLAB and specific MEMS test routines.

As a result of the creation of this toolbox, it was possible to provide the user with a simpler more intuitive way to interact with the MEMS test platform [2], adding the reliability of a driver to perform the communication and expanding the features of the system, in terms of acquisition and test modes.

Keywords: Toolbox, MEMS test, GUI, Driver, MATLAB



# Conteúdo

Agradecimentos .....	iii
Resumo .....	v
Abstract .....	vii
1. Introdução.....	1
1.1. Enquadramento .....	2
1.2. Motivação .....	3
1.3. Objetivos .....	4
1.4. Estrutura da Dissertação.....	5
2. Estado da Arte .....	7
2.1. Estruturas MEMS.....	8
2.1.1. Acelerómetros electroestáticos.....	9
2.2. Fases de Teste.....	10
2.3. Tipos de Teste.....	11
2.3.1. Análise ótica por vibração.....	11
2.3.2. Sistema de teste mecânico baseado em pontas de prova .....	12
2.3.3. Medição da resposta a estímulos elétricos .....	13
2.4. ITMEMS MCP.....	14
2.4.1. Capacidades analíticas.....	14
2.4.2. Especificações técnicas.....	15
2.5. Embedded MEMS Platform for structure test and characterization .....	16
2.6. Conclusão do Estado da Arte .....	16
3. Plataforma de teste de MEMS .....	17
3.1. Visão geral do sistema.....	18
3.1.1. Sistema de Leitura.....	19

3.1.2.	Sistema de Atuação .....	19
3.2.	Hardware programável (FPGA).....	20
3.3.	Software do microprocessador.....	21
3.3.1.	FreeRTOS .....	21
3.3.2.	lwIP .....	21
3.3.3.	Funcionamento do sistema.....	22
3.3.4.	Mapeamento dos Registos AXI .....	26
3.3.5.	Protocolo de Comunicação.....	27
3.4.	Driver de comunicação.....	28
3.5.	Interface Gráfica.....	29
3.6.	Conclusão do capítulo.....	29
4.	Funcionalidades da Toolbox.....	31
4.1.	Visão geral do sistema.....	32
4.2.	Painel Read Channel.....	37
4.2.1.	Configuração AC/DC.....	38
4.2.2.	Configuração Amplificador de Carga .....	40
4.2.3.	Configuração Single Ended/Diferencial.....	42
4.2.4.	Configuração do ganho de entrada.....	43
4.2.5.	Configuração do ADC.....	44
4.3.	Painel DAC.....	47
4.3.1.	Seleção do modo de funcionamento do DAC.....	48
4.3.2.	Configuração do Wave Generator .....	50
4.3.3.	Interruptores Analógicos.....	57
4.4.	Painel Lock-In .....	58
4.4.1.	Configuração da frequência.....	59
4.4.2.	Configuração do ganho .....	60

4.4.3.	Configuração do filtro decimador .....	61
4.4.4.	Sintetização de filtros.....	62
4.5.	Painel Acquire.....	65
4.5.1.	Seleção da frequência de amostragem .....	67
4.6.	Painel Live Mode.....	68
4.7.	Controlos de visualização.....	70
4.8.	Painel de configuração da comunicação.....	71
4.9.	Painel Pull-in .....	71
4.10.	Painel Frequency Sweep .....	73
4.11.	Conclusão do capítulo .....	74
5.	Resultados.....	75
5.1.	Resultados das configurações .....	76
5.2.	Resultados dos blocos de aquisição .....	78
5.2.1.	Acquire Mode .....	78
5.2.2.	Live Mode.....	80
5.3.	Resultados das rotinas de teste de MEMS .....	81
5.3.1.	Tensões de pull-in e pull-out.....	82
5.3.2.	Frequency Sweep.....	83
6.	Conclusões e Trabalho Futuro .....	87
	Referências Bibliográficas .....	89
	Anexos.....	91
	A: Mensagens de Comunicação .....	92





# Lista de Acrónimos

A – Ampere

AC – *Alternating Current*

ADC – *Analog to Digital Converter*

AMBA – *Advanced Microcontroller Bus Architecture*

ARM – *Advanced RISC Machine*

AXI – *Advanced eXtensible Interface*

CI – *Circuito integrado*

CMOS – *Complementary Metal-Oxide-Semiconductor*

DAC – *Digital to Analog Converter*

DC – *Direct Current*

DDS – *Digital Direct Synthesizer*

DUT – *Device Under Test*

DVGA – *Digital Variable Gain Amplifier*

dB – *Decibel*

F – *Farad*

FPGA – *Field-Programmable Gate Array*

GPIO – *General Purpose Interface Bus*

GUI – *Graphical User Interface*

Hz – *Hertz*

I – *In-phase Component*

IoT – *Internet of Things*

IP – *Intellectual Property*

KGB – *Known Good Dies*

LASER – *Light Amplification by Stimulated Emission of Radiation*

LwIP – *Lightweight Internet Protocol*

MATLAB – *MATrix LABoratory*

MEMS – *Microelectromechanical Systems*

MCP – *MEMS Characterization Platform*

PCB – *Printed Circuit Board*

PWM – *Pulse Width Modulation*

Q – *Quadrature Component*

RAM – *Random Access Memory*

ROM – *Read-Only Memory*

RISC – *Reduced Instruction Set Computer*

s – *Segundos*

SoC – *System on Chip*

TCP/IP – *Transmission Control Protocol / Internet Protocol*

TTL – *Transistor-Transistor Logic*

V – *Volt*

VLSI – *Very Large Scale Integration*

WSN – *Wireless Sensor Networks*

# Lista de Figuras

Figura 1-1 - Previsão do mercado de MEMS [3].	2
Figura 2-1 - Imagem SEM de uma estrutura MEMS [7].	8
Figura 2-2 - Princípio de funcionamento de um acelerómetro MEMS capacitivo [5].	9
Figura 2-3 MSA-500 - Micro-System Analyzer da Polytec Inc [10].	12
Figura 2-4 - FT-MPS02 Probestation da FEMTOTools [11].	13
Figura 2-5 - ITMEMS MCP [12].	14
Figura 2-6 - Arquitetura do sistema de teste de MEMS N. Brito et. al. [7].	16
Figura 3-1 - Visão geral da plataforma de teste de MEMS.	18
Figura 3-2 - Arquitetura do hardware programável [2].	20
Figura 3-3 - Fluxogramas das tarefas Main e Run previamente implementadas [2].	22
Figura 3-4 - Fluxograma da tarefa Main desenvolvida.	23
Figura 3-5 - Fluxograma da tarefa ProcessRequest desenvolvida.	24
Figura 3-6 - Fluxograma da tarefa SendingTask desenvolvida.	25
Figura 3-7 - Registos de escrita da plataforma de teste de MEMS.	26
Figura 3-8 - Registos de leitura da plataforma de teste de MEMS.	27
Figura 3-9 - Visão geral do funcionamento do MATLAB Instrument Driver [20].	28
Figura 4-1 - Painel principal da interface gráfica.	32
Figura 4-2 - Fluxograma do método set.	34
Figura 4-3 - Fluxograma do método get.	35
Figura 4-4 - Fluxograma do método invoke.	36
Figura 4-5 - Painel Read Channel Configuration.	37
Figura 4-6 - Arquitetura do sistema de aquisição da plataforma de teste de MEMS[2].	37
Figura 4-7 - Painel AC_DC do separador Read Channel.	38
Figura 4-8 - Painel Charge_Direct do separador Read Channel.	40
Figura 4-9 - Painel Single_Differential do separador Read Channel.	42
Figura 4-10 - Painel Gain_IN Configuration do separador Read Channel.	43

Figura 4-11 - Painel ADC Configuration do separador Read Channel.....	45
Figura 4-12 - Arquitetura do sistema de atuação da plataforma de teste de MEMS [2] .....	47
Figura 4-13 Painel Select do separador DAC. ....	48
Figura 4-14 - Painel Wave Generator Configuration do separador DAC.....	50
Figura 4-15 - Painel Type do separador Wave Generator. ....	50
Figura 4-16 - Painel Frequency do separador Wave Generator.....	52
Figura 4-17 - Painel Amplitude do separador Wave Generator.....	53
Figura 4-18 - Painel Phase do separador Wave Generator.....	54
Figura 4-19 - Painel Offset do separador Wave Generator.....	55
Figura 4-20 - Painel Duty Cycle do separador Wave Generator.....	56
Figura 4-21 - Painel dos interruptores analógicos. ....	57
Figura 4-22 - Arquitetura do amplificador de lock-in. ....	58
Figura 4-23 - Painel Lock-in Configuration. ....	58
Figura 4-24 - Painel “Frequency” do separador Lock-in. ....	59
Figura 4-25 - Painel Gain do separador Lock-in. ....	60
Figura 4-26 - Painel Decimator do separador Lock-in.....	61
Figura 4-27 - Painel Synthesize do separador Lock-in.....	62
Figura 4-28 - Exemplo de janelas Filter Visualization Tool para os quatro tipos de filtro.....	63
Figura 4-29 - Painel Manual Input do separador Lock-in. ....	63
Figura 4-30 - Painel Acquire.....	65
Figura 4-31 - Dial de seleção do modo de aquisição. ....	66
Figura 4-32 - Painel de configuração da frequência de amostragem. ....	67
Figura 4-33 - Painel Live Mode.....	68
Figura 4-34 - Botão Live Mode e dial Display Mode.....	68
Figura 4-35 - Painel de controlos manuais de visualização. ....	70
Figura 4-36 - Painel de configuração da comunicação.....	71
Figura 4-37 - Painel Pull-in. ....	72
Figura 4-38 - Painel Frequency Sweep.....	73
Figura 5-1 - Resultados da configuração do filtro decimador do amplificador de lock- in.....	76

Figura 5-2 - Resultados da configuração do ganho do amplificador de lock-in. ....	77
Figura 5-3 - Resultado da configuração do gerador de ondas com uma onda triangular. ....	77
Figura 5-4 - Resultados do modo de aquisição para diferentes durações.....	79
Figura 5-5 - Resultados do funcionamento do "Live Mode" e "Acquire Mode". ....	80
Figura 5-6 - Resultados da caracterização de acelerômetros MEMS capacitivos da Embedded MEMS Platform for structure test and characterization [7].....	81
Figura 5-7 - Resultados da medição da tensão de pull-in e pull-out na GUI. ....	82
Figura 5-8 - Gráficos dos resultados da medição da tensão de pull-in e pull-out....	83
Figura 5-9 - Resultados do varrimento de frequências na GUI.....	84
Figura 5-10 - Gráfico dos resultados do varrimento de frequências.....	85
Figura A.1 - Mensagens de comunicação. ....	92



# Lista de Tabelas

Tabela 4-1 - Métodos AC_DC.....	39
Tabela 4-2 - Valores permitidos para a configuração do AC_DC.....	39
Tabela 4-3 - Métodos Charge_Direct.....	40
Tabela 4-4 - Valores permitidos para a configuração do Charge_Direct.....	41
Tabela 4-5 - Métodos Charge_Gain.....	41
Tabela 4-6 - Valores permitidos para a configuração do Charge_Gain.....	42
Tabela 4-7 - Métodos Single_Differential.....	43
Tabela 4-8 - Valores permitidos para a configuração do Single_Differential.....	43
Tabela 4-9 - Métodos Gain_In.....	44
Tabela 4-10 - Métodos ADC_Gain.....	45
Tabela 4-11 - Valores permitidos para a configuração do ADC_Gain.....	46
Tabela 4-12 - Métodos ADC_Freq.....	46
Tabela 4-13 - Métodos DAC_Select.....	49
Tabela 4-14 - Valores permitidos para a configuração do DAC_Select.....	49
Tabela 4-15 - Métodos WG_Type.....	51
Tabela 4-16 - Valores permitidos para a configuração do WG_Type.....	51
Tabela 4-17 - Métodos WG_Freq.....	52
Tabela 4-18 - Métodos WG_Amp.....	53
Tabela 4-19 - Métodos WG_Phase.....	54
Tabela 4-20 - Métodos WG_Offset.....	55
Tabela 4-21 - Métodos WG_Duty.....	56
Tabela 4-22 - Métodos Switch.....	57
Tabela 4-23 - Métodos LI_Freq.....	59
Tabela 4-24 - Métodos LI_Gain.....	60
Tabela 4-25 - Métodos LI_Dec.....	61
Tabela 4-26 - Métodos de configuração dos coeficientes.....	64
Tabela 4-27- Métodos do modo Acquire.....	66
Tabela 4-28 - Valores para seleção do modo de aquisição.....	66
Tabela 4-29 - Métodos Sampling Frequency.....	67

## Lista de Tabelas

---

Tabela 4-30-Métodos Live Mode. ....	69
Tabela 4-31 - Valores para seleção do modo de aquisição do Live Mode.....	69
Tabela 4-32 - Métodos da rotina de medição da tensão de pull-in.....	72
Tabela 4-33 - Métodos da rotina de varrimento de frequências.....	74



# 1.

# Introdução

Este capítulo é iniciado com uma abordagem ao enquadramento do tema desta dissertação, seguindo-se uma apreciação do que motivou a elaboração da mesma. São ainda elencados os objetivos que se espera obter com a conclusão deste projeto. Por fim, é feita uma análise à estrutura definida para a escrita deste documento.

## 1.1. Enquadramento

Desde a década de 70, os sistemas microeletromecânicos (MEMS) têm emergido com sucesso baseando-se na existente e bem estabelecida indústria dos circuitos integrados (IC).

Com a atual desaceleração do crescimento do mercado dos *smartphones* e outros dispositivos portáteis, seria de esperar uma estabilização do mercado de MEMS a longo prazo, no entanto, existem outros mercados emergentes, nomeadamente, WSN (*Wireless Sensor Networks*) e IoT (*Internet of Things*), que segundo analistas da Yole, poderão impulsionar este crescimento nos próximos anos [3], como pode ser visualizado na Figura 1-1.

Mantendo-se esta tendência, torna-se crucial desenvolver tecnologias que permitam melhorar o processo de *design*, desenvolvimento, fabrico e teste destes dispositivos.

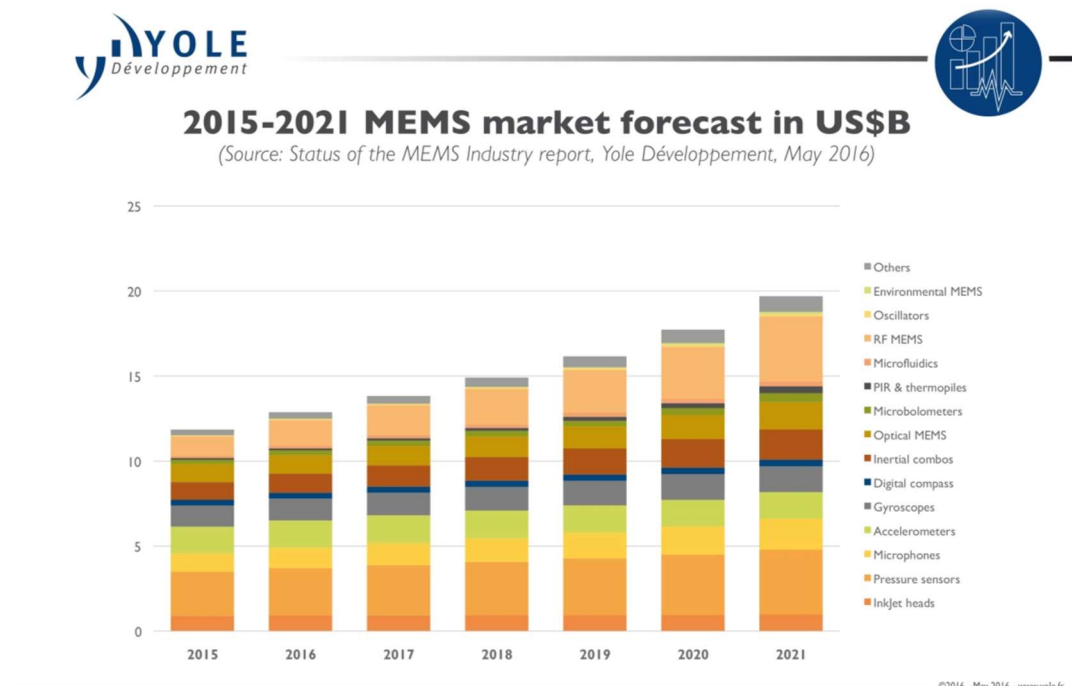


Figura 1-1 - Previsão do mercado de MEMS [3].

## 1.2. Motivação

Dispositivos MEMS são sensores e atuadores eletromecânicos fabricados utilizando as mesmas técnicas e processos empregues na produção de microeletrónica. No entanto, devido à sua natureza mecânica, a complexidade do processo de teste de MEMS é consideravelmente superior [4].

Atualmente, existe uma falta de sistemas de teste de MEMS disponíveis no mercado, principalmente sistemas rápidos e de baixo custo que possam ser utilizados tanto em ambiente laboratorial, como em processo fabril. Sendo estes importantes no desenvolvimento de microestruturas, quer para a sua caracterização no processo de fabrico, quer para a sua calibração em processo fabril, é de especial interesse a criação de uma plataforma de testes deste tipo.

Com o intuito de solucionar este problema, foi iniciado o processo de desenvolvimento de um destes sistemas, na Universidade do Minho. O *hardware* desta plataforma já foi criado utilizando uma FPGA e placas de circuito impresso, sendo que o próximo passo é desenvolver o *software* que permita que esta efetue as principais rotinas de teste de MEMS e que possibilite uma forma simples de comunicação entre o utilizador e a plataforma [2].

Com o desenvolvimento desta dissertação, um projetista de MEMS poderá efetuar os principais testes de caracterização das suas estruturas de uma forma simples e intuitiva, sem a necessidade de conhecer a arquitetura da plataforma de testes e sem necessitar de programar os vários componentes, incluindo a FPGA, do sistema de testes.

## 1.3. Objetivos

A finalidade desta dissertação é o desenvolvimento de uma *toolbox* que permita expandir as funcionalidades da plataforma de teste de MEMS [2].

De forma a alcançar esse objetivo, a elaboração desta *toolbox* pode ser dividida nas seguintes três componentes e respectivas subcomponentes:

- *Driver* de comunicação com a plataforma de teste MEMS:
  - Interpretação e tradução do protocolo de comunicação.
- Interface gráfica:
  - Desenvolvida em MATLAB;
  - Deve permitir configurar as propriedades da plataforma de teste;
  - Deve possibilitar a aquisição dos sinais e valores de saída desta.
- Rotinas de teste de MEMS capacitivos em *software*:
  - *Live Mode*;
  - Varrimento de frequências;
  - Medição de tensões de *pull-in* e *pull-out*;
  - Medição da frequência de ressonância;
  - Medição do fator de qualidade;
  - Aquisição do diagrama de Bode;
  - Medição da capacidade em repouso.

## 1.4. Estrutura da Dissertação

O primeiro capítulo desta dissertação é de caráter introdutório, sendo abordada a motivação e os objetivos desta dissertação.

No segundo capítulo, é feita uma análise ao estado da arte relativo ao teste de MEMS, referenciando alguns sistemas relevantes para o desenvolvimento desta *toolbox*.

A plataforma de teste de MEMS utilizada como base desta dissertação é apresentada no terceiro capítulo, descrevendo algumas das suas características principais e os mecanismos de funcionamento da mesma, explicando as alterações efetuadas ao *hardware* e *software* existentes.

O quarto capítulo descreve as funcionalidades da *toolbox* desenvolvida, explicando como utilizar cada uma destas e esmiuçando o seu funcionamento e finalidade.

Os resultados obtidos com o desenvolvimento desta *toolbox* são expostos e analisados no quinto capítulo deste documento.

Por fim, o sexto capítulo é dedicado às conclusões obtidas durante o desenvolvimento desta dissertação, apresentando recomendações para trabalho futuro de forma a melhorar os resultados obtidos e a expandir as funcionalidades do sistema elaborado.



# 2.

## Estado da Arte

Dispositivos MEMS são normalmente fabricados utilizando técnicas de fabrico compatíveis com as dos VLSI IC (*Very Large Scale Integration Intergrated Circuit*), no entanto, os métodos de teste de MEMS são consideravelmente distintos dos utilizados em circuitos VLSI. Devido à sua natureza mecânica, as estruturas MEMS necessitam de considerações especiais durante o processo de teste nas fases de desenho, fabrico e calibração. O tempo de teste torna-se também uma característica absolutamente crítica nas fases de fabrico e calibração [2]. Estes métodos de teste variam também consoante a fase de fabrico em que são executados, ou seja, se o teste é efetuado antes do encapsulamento na *wafers* de silício ou depois de o dispositivo ser encapsulado [2].

## 2.1. Estruturas MEMS

A evolução das técnicas de micromaquinagem permitiram uma redução do tamanho dos sensores mecânicos e um aumento das suas funcionalidades a níveis sem precedentes de miniaturização, resultando destes avanços o desenvolvimento de sistemas MEMS [5], tal como o apresentado na Figura 2-1. Estes dispositivos consistem em pequenos sistemas integrados que combinam componentes mecânicas e elétricas, podendo as suas dimensões variar na ordem dos nanómetros a alguns milímetros. Atualmente, estes dispositivos podem ser encontrados nas mais variadas áreas, tais como, automóvel, médica, eletrónica, comunicações e militar. Alguns exemplos destas aplicações são acelerómetros para airbags e smartphones, tinteiros para impressoras a jato de tinta, micro espelhos para projetores de imagem, entre outros [6]. Existe uma imensa variedade de transdutores mecânicos que podem ser replicados em sistemas MEMS, sendo estes habitualmente piezoresistivos, nos quais a resistividade do seu material varia em função da deformação, piezoelétricos, em que o potencial elétrico de um cristal piezoelétrico varia ao longo do cristal em função de uma força ou deformação, e podem ainda ser capacitivos, nos quais a capacidade dos condensadores varia em função do seu deslocamento em relação a uma referência, sendo estes os mais comuns.

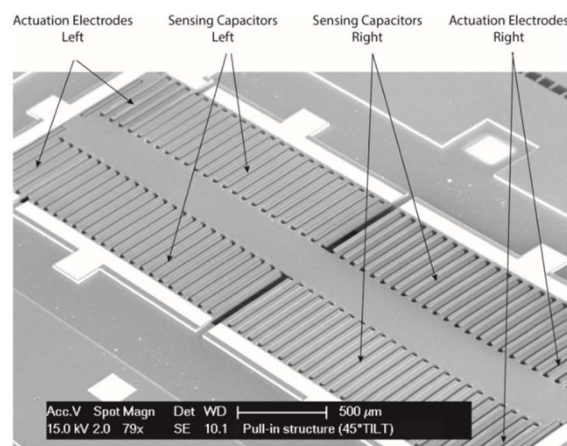


Figura 2-1 - Imagem SEM de uma estrutura MEMS [7].



### 2.1.1. Acelerómetros capacitivos

Um exemplo deste tipo de sistemas, amplamente utilizado nas mais variadas áreas, é o acelerómetro capacitivo, graças à sua elevada precisão, linearidade e estabilidade térmica. O funcionamento deste tipo de sensor, demonstrado na Figura 2-2, consiste num sistema massa-mola, com a massa de prova ligada a uma referência através de molas. Quando sujeito a acelerações ao longo do eixo de leitura, ocorre a deflexão desta massa, sendo estas forças externas balanceadas pela força elástica das molas. Em MEMS capacitivos *parallel-plate* (elétrodos paralelos), este movimento de deflexão da massa de prova é perpendicular ao plano das placas do condensador formado entre a massa de prova e os elétrodos fixos. Este movimento leva a uma alteração da posição relativa entre a massa e os elementos capacitivos, traduzindo-se numa variação da capacidade. Neste tipo de dispositivo, as placas de condensadores podem ainda servir como elementos atuadores, com a aplicação de uma tensão na estrutura capacitiva, ficando a estrutura sujeita a uma força electrostática, resultando no movimento da massa de prova. Esta característica é uma das grandes vantagens deste tipo de MEMS, permitindo implementar mecanismos de autoteste para caracterização dos mesmos [5].

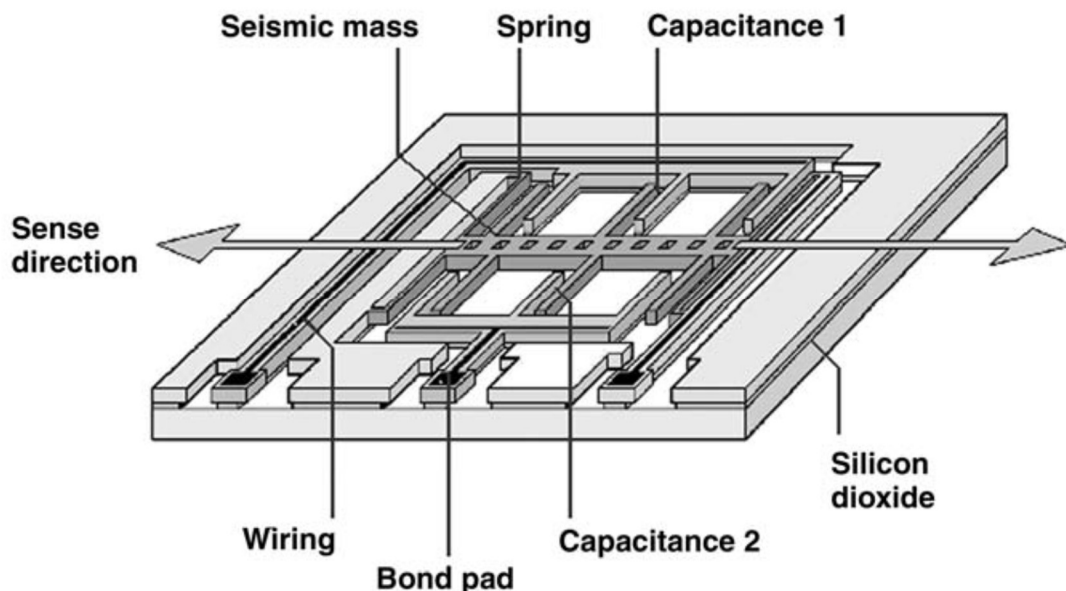


Figura 2-2 - Princípio de funcionamento de um acelerómetro MEMS capacitivo [5].

## 2.2. Fases de Teste

Testes ao nível da *wafer* requerem a utilização de pontas de prova controladas com alta precisão para verificar individualmente o correto funcionamento de cada *die* da *wafer*. Testar este tipo de dispositivos está normalmente limitado a teste de teor elétrico e, embora permitam retirar informações básicas sobre o funcionamento de cada *die*, não permitem analisar as características mecânicas destes sensores e atuadores. Tal como nos sistemas VLSI, proceder a uma análise ao nível da *wafer* em MEMS torna possível melhorar o rendimento do processo de fabrico e reduzir os custos de produção. Para novos dispositivos que estejam em fase de *design* e desenvolvimento, testes desta natureza permitem efetuar uma avaliação mais detalhada do produto e, conseqüentemente, validar o funcionamento do mesmo. Para tecnologias MEMS já maduras, esta abordagem possibilita a identificação das *known-good-dies* (KGD), ou seja, identificar as *dies* da *wafer* que estão a funcionar corretamente, para selecionar quais destas devem passar para a fase de empacotamento. Estudos revelam que mais de 60% dos custos de produção se devem ao processo de empacotamento. Desta forma, retirando da equação o empacotamento das chamadas “*bad dies*”, é proporcionada uma considerável redução dos custos de produção [4][8].

Dispositivos MEMS já empacotados podem ser testados aplicando estímulos de qualquer natureza às suas entradas, dependendo do tipo de sistema MEMS. Para este fim, podem ser utilizados muitos dos métodos normalmente aplicados no teste de ICs já empacotados, por ex., ciclos térmicos, armazenamento em altas temperaturas, choque térmico e alta humidade [4]; porém, no teste de MEMS, devido à presença de partes móveis na sua composição, é fundamental efetuar uma gama acrescida de testes, para avaliar a influência da humidade, pressão, gases presentes no ambiente e contaminação por partículas, no movimento dessas componentes.

## **2.3. Tipos de Teste**

Sistemas MEMS podem ser utilizados em diversas áreas de aplicação devido à grande variedade de tipos de dispositivos disponíveis atualmente, desde inerciais (por ex., acelerômetros e giroscópios), radiofrequência, entre outros. Esta diversidade implica uma maior especificidade quanto ao tipo de testes a ser aplicados para determinado tipo de sensor, dada a necessidade da criação de estímulos externos adequados ao tipo de sensor em causa. Embora não exista atualmente uma solução “one-size fits all” [9], é possível testar estes sistemas através de três métodos distintos, análise ótica por vibração, análise mecânica e medição da resposta a estímulos elétricos.

### **2.3.1. Análise ótica por vibração**

Tipicamente, os dispositivos MEMS têm uma componente mecânica associada aos sinais elétricos. Esta característica permite que, através de inspeção ótica, o sistema seja caracterizado baseando-se no seu movimento mecânico. Utilizando um detetor de movimento estroboscópico ou um vibrómetro a LASER (por efeito de Doppler), a velocidade de vibração e o deslocamento do dispositivo MEMS podem ser medidos.

Este tipo de teste tem como principal vantagem tornar prescindível o contacto direto com o dispositivo sob teste[2]. Um exemplo comercial deste tipo de sistema de teste é o MAS-500 Micro System Analyzer, apresentado na Figura 2-3 e comercializado pela Polytec Inc [10].



Figura 2-3 MSA-500 - Micro-System Analyzer da Polytec Inc [10].

### **2.3.2. Sistema de teste mecânico baseado em pontas de prova**

Neste tipo de teste podem ser utilizadas várias abordagens distintas quanto ao método de aplicação do estímulo mecânico no dispositivo sob teste. Estes baseiam-se em aplicar uma translação, vibração, rotação ou choque mecânico e avaliar a resposta do sensor ou atuador MEMS [4].

A empresa FEMTO Tools possui um sistema de teste deste tipo, o FT-MPS02 MEMS Probestation [11], mostrado na Figura 2-4.

Utilizando nano-posicionamento e micro-pontas de prova que aplicam uma força e medem a sua resposta, este produto permite caracterizar o dispositivo em teste. Esta abordagem tem como desvantagem o tempo necessário para efetuar o teste e a necessidade das estruturas sob teste estarem diretamente acessíveis às micro-pontas de prova [2].

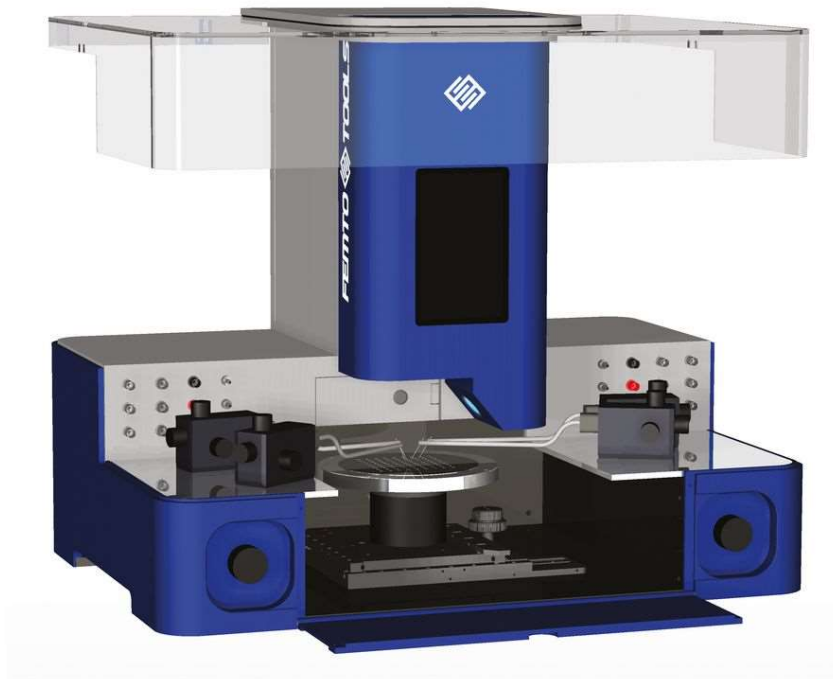


Figura 2-4 - FT-MPS02 Probestation da FEMTOTools [11].

### 2.3.3. Medição da resposta a estímulos elétricos

O teste elétrico de MEMS é um dos métodos mais importantes utilizados para caracterizar MEMS e nasce a partir dos métodos tradicionais utilizados no teste de circuitos integrados [4]. Utilizando micro-pontas de prova para injeção de sinais e medição da resposta do sistema, este tipo de teste permite caracterizar o dispositivo em teste sem a necessidade de estímulos mecânicos externos, ou seja, aplicando à estrutura um estímulo elétrico e medindo a sua saída elétrica, é possível utilizar esta informação para obter as características mecânicas da estrutura. Esta abordagem tem como vantagens o facto de ser rápida e flexível, permitindo a sua utilização nas três fases de desenvolvimento das estruturas MEMS supracitadas[2].

## 2.4. ITMEMS MCP

A *MEMS Characterization Platform* da ITMEMS [12], apresentada na Figura 2-5, é uma plataforma de caracterização de sensores MEMS capacitivos, *parallel-plate* e *comb-finger*, disponível comercialmente e que permite a caracterização das propriedades elétricas e mecânicas de dispositivos capacitivos com frequências de ressonância até 500 kHz.

Inicialmente desenvolvida para acelerômetros e giroscópios, esta pode ainda ser utilizada para testar magnetômetros, micro-espelhos, microfones, sensores de pressão, etc. Valendo-se de uma interface gráfica para PC e comunicação USB, esta plataforma é direcionada apenas a testes laboratoriais, dada a baixa velocidade de execução de testes.



Figura 2-5 - ITMEMS MCP [12].

### 2.4.1. Capacidades analíticas

- Modos de medição (*single ended* e diferencial):
  - Capacidade em repouso;
  - Variação da capacidade estacionária;
  - Variação da capacidade dinâmica sob aplicação de estímulos electrostáticos (sinusoides, ondas quadradas, triangulares, etc.);
  - Variação da capacidade dinâmica sob forças externas sincronizadas (tensões, acelerações, velocidade angular, etc.);
  - Varrimento de frequências.

- Extração direta de parâmetros (sem *input* do utilizador):
  - *Offset* capacitivo (em MEMS com configurações diferenciais);
  - Tensões de *pull-in* e *pull-out*;
  - Frequência de ressonância;
  - Fator de qualidade;
  - Diagramas de Bode;
  - Largura de banda 3 dB em MEMS sobre amortecidos;
  - Efeitos de amortecimento electroestático;
  - Não-linearidades mecânicas.
- Medições indiretas (com *input* fornecido pelo utilizador):
  - *Offset* mecânico (em MEMS com configurações diferenciais);
  - *Surface charging*;
  - *Over/under etching* do processo de fabrico;
  - Erro de quadratura em giroscópios;
  - Módulo de Young e força de fratura;
  - Fadiga e outros testes de fiabilidade;
  - Forças de adesão e deterioração da superfície de contacto.

## 2.4.2. Especificações técnicas

- Interface com o dispositivo MEMS:
  - Capacidades até 10 pF;
  - Frequências de ressonância até 500 kHz.
- Sinais de alta frequência:
  - Sinusoides até 1 MHz para MCP-A;
  - Sinusoides até 5 MHz para MCP-G;
  - Amplitudes entre 10 mV e 700 mV.
- Resolução:
  - 0,3%/√Hz da capacidade medida entre 200 fF e 10 pF, para sinais de teste a 50 mV.
- Tensões de atuação:

Sinais estacionários e dinâmicos até ±13 V (MCP-A) e ±23 V (MCP-G).

## 2.5. Embedded MEMS Platform for structure test and characterization

Outro exemplo de sistemas de caracterização que utilizam este método é a plataforma desenvolvida *circa* 2015, por N. Brito *et. al* [7]. Este produto não comercial permite a execução dos seguintes testes:

- Curvas de tensão de *pull-in*;
- Frequência de ressonância;
- Fator de qualidade.

Este sistema, mais adequado à fase de produção industrial, permite efetuar todos estes testes em 1,5 segundos, com uma precisão de 5 ppm, utilizando a arquitetura presente na Figura 2-6.

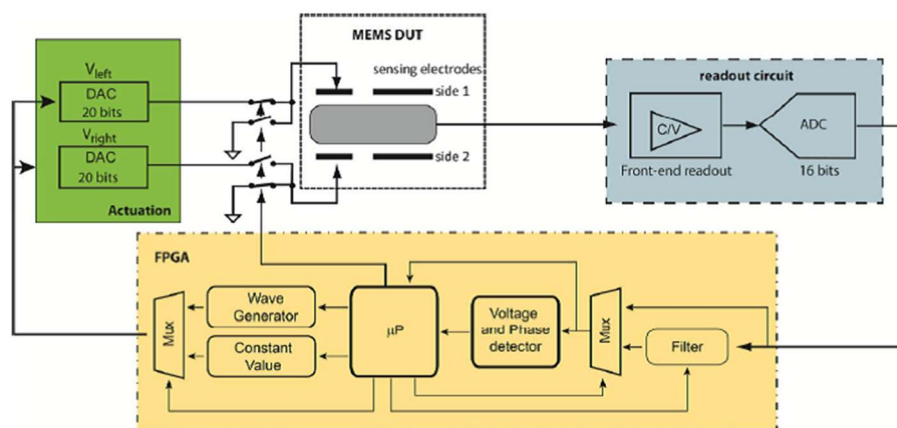


Figura 2-6 - Arquitetura do sistema de teste de MEMS N. Brito *et. al.* [7].

## 2.6. Conclusão do Estado da Arte

Neste estado de arte, é apresentada uma apreciação dos vários tipos de teste de MEMS e sistemas de teste disponíveis para fins laboratoriais e comerciais. Os blocos de *hardware* desta já foram desenvolvidos, no entanto, comparando com o sistema da ITMEMS, esta carece de uma interface gráfica de comunicação e de rotinas específicas de teste de MEMS presentes na ITMEMS MCP. Tendo como principais vantagens a inclusão de um amplificador de *lock-in* digital programável e conversores capacidade/tensão, torna-se numa solução adequada para ambiente laboratorial e industrial.



# 3.

## Plataforma de teste de MEMS

Sendo o objetivo desta dissertação a criação de uma *toolbox* para a plataforma de teste de MEMS desenvolvida na Universidade do Minho em 2016, é efetuada, neste capítulo, uma descrição deste sistema, especificando para cada bloco as alterações efetuadas durante o desenvolvimento desta *toolbox*.

No primeiro subcapítulo, é apresentada uma visão geral deste sistema, abordando a sua arquitetura e a forma como os vários blocos que a compõem se relacionam. Os dois subcapítulos que o sucedem fazem uma análise do *hardware* programável e do *software* do microprocessador, respetivamente, seguindo-se, no subcapítulo 3.4, a descrição do *driver* de comunicação desenvolvida e o *software* MATLAB Instrument Driver utilizado no seu desenvolvimento. Este capítulo é finalizado com a exposição da interface gráfica criada com o *software* App Designer.

### 3.1. Visão geral do sistema

A plataforma de teste de MEMS utilizada como base desta dissertação é um sistema integrado de teste e caracterização para sistemas microeletromecânicos com o objetivo de tornar mais simples e rápida a análise deste tipo de estruturas, tendo como principais características a inclusão de um amplificador *lock-in* digital programável e um conversor capacidade/tensão com ganho variável.

Esta plataforma, como é possível evidenciar na Figura 3-1, pode ser dividida em três camadas principais, o *hardware* de leitura e atuação, o *hardware* programável (FPGA) e o *software* do microprocessador. Esta figura especifica também as camadas da plataforma onde esta dissertação incide, nomeadamente, a FPGA e o *software* implementado no ARM, assim como os blocos, *Instrument Driver* e interface gráfica, desenvolvidos no âmbito desta *toolbox*.

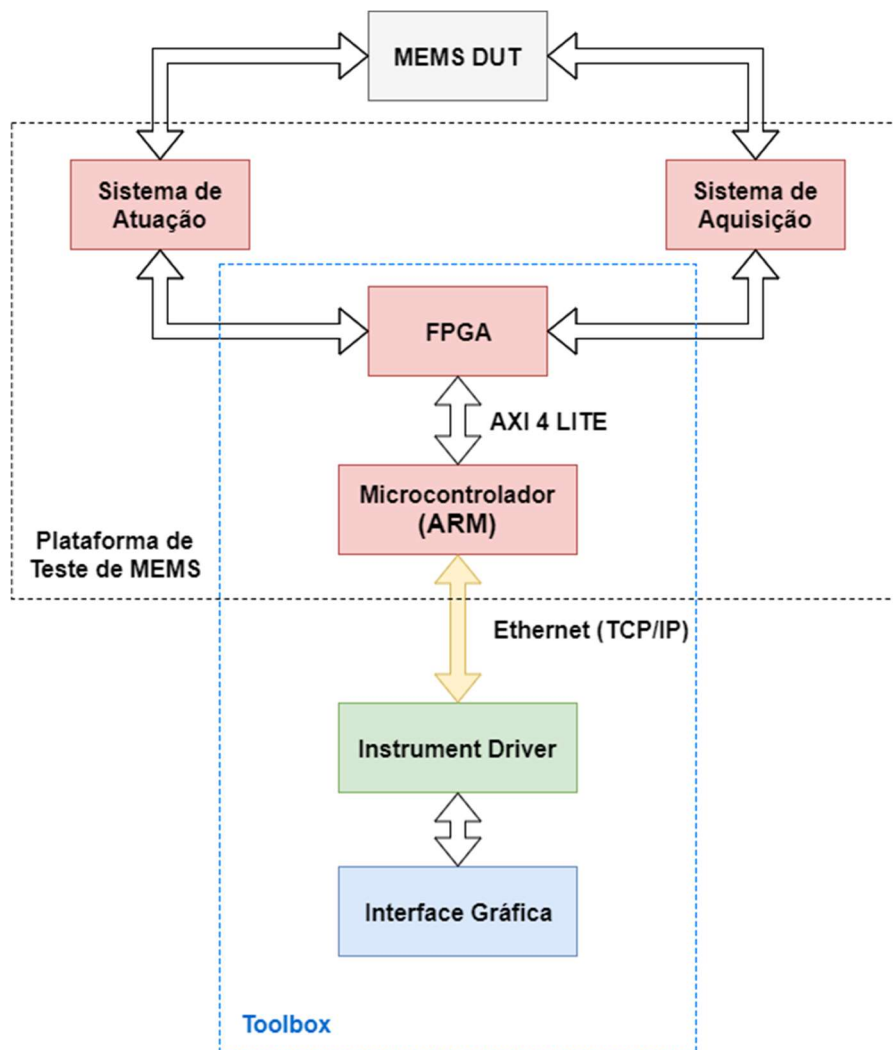


Figura 3-1 - Visão geral da plataforma de teste de MEMS.

### 3.1.1. Sistema de Leitura

O sistema de aquisição conta com um amplificador de carga de ganho variável, conversão de sinais *single ended* para diferencial, um amplificador de ganho variável e um conversor analógico-digital. As suas características principais são as seguintes:

- Canal de Entrada
  - Largura de Banda: DC-5 MHz;
  - Impedância de Entrada: 50  $\Omega$ ;
  - Ruído Tensão de Entrada: 114,2 nV/ $\sqrt{\text{Hz}}$ ;
  - Excursão de entrada:  $\pm 1,149 \mu\text{V}$  a  $\pm 4 \text{ V}$ ;
  - Conversão A/D: 16 bits, 17,5 MS/s;
  - Ganho Variável: -6 dB a 32 dB.

### 3.1.2. Sistema de Atuação

De forma a atuar os dispositivos MEMS a testar, a plataforma inclui quatro conversores digital-analógico, quatro interruptores e um DDS (*Digital Direct Synthesizer*) para geração das portadoras. Este tem como características:

- Geração de Portadoras
  - Frequência: 1 Hz a 10 MHz;
  - Excursão de Saída: 50 mV a 1,5 V (Intervalos de 1 dB);
  - Conversor A/D: 10 bits / 105 MHz.
- Interruptores Analógicos:
  - Excursão de saída:  $\pm 15 \text{ V}$ ;
  - Corrente Máxima: 115 mA;
  - Entradas: 3 V TTL;
  - Tempo de Comutação: 140 nS.
- 4x Conversores D/A
  - Frequência de Amostragem: 1,4 MHz;
  - Resolução: 20 bits;
  - LSB:  $\pm 19,073 \mu\text{V}$ ;
  - Excursão de saída:  $\pm 10 \text{ V}$ ;
  - Sinais: Amplitude, Fase, ADC e Geração de Ondas;
  - Tipos de Onda: Sinusoide, Triangular, Quadrada, PWM e Rampa.

Estes elementos são novamente abordados no capítulo 4, na descrição dos seus painéis de configuração na GUI. Os blocos restantes abrangidos por esta *toolbox* são descritos nas subsecções posteriores.

### 3.2. Hardware programável (FPGA)

De modo a adquirir e processar os sinais adquiridos na primeira camada deste sistema de testes com uma frequência de amostragem elevada, gerir a comunicação entre periféricos em *hardware* e garantir a execução de funções em paralelo, é utilizada, nesta segunda camada, uma FPGA.

A placa de desenvolvimento utilizada nesta plataforma é a ZedBoard [13], que contém o SoC XC7Z020, da família Zynq 7000 da Xilinx. Este SoC (System On Chip), possui uma FPGA da família Artix-7 da Xilinx com cerca de 85.000 células programáveis e 220 células DSP [14].

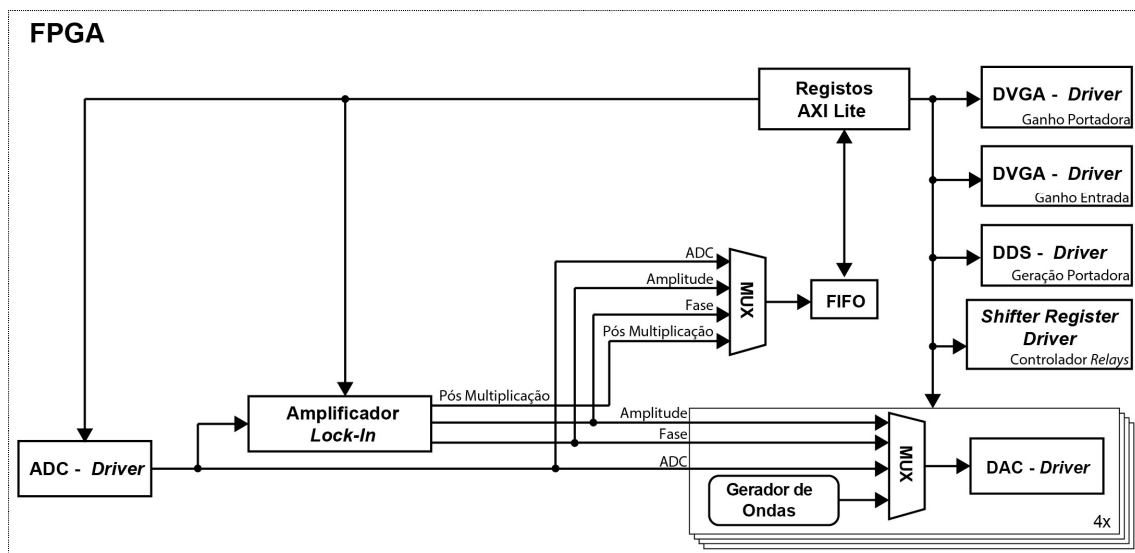


Figura 3-2 - Arquitetura do hardware programável [2].

A arquitetura deste hardware programável, que pode ser visualizada na Figura 3-2, não foi alterada no desenvolvimento desta dissertação, tendo apenas sido efetuados ajustes ao nível de alguns blocos IP (Intellectual Property) de modo a garantir o correto funcionamento da *toolbox*.

Para alterar estes módulos, implementados em Verilog, foi utilizado o ambiente de desenvolvimento e simulação Vivado Suite da Xilinx.

## 3.3. Software do microprocessador

A terceira camada deste sistema é o ARM no mesmo chip da FPGA, sendo este responsável por, através do LwIP, gerir a comunicação TCP/IP entre a plataforma e o cliente. Permite também configurar a FPGA de acordo com o requerido pelo utilizador, servindo-se dos registos AXI. De maneira a manter este balanço entre configurações e transferência de dados, e garantindo que as restrições temporais do sistema são mantidas, é utilizado o sistema operativo de tempo real FreeRTOS.

O SoC incluído na placa de desenvolvimento Zedboard contém dois ARM Cortex-A9 *dual-core* de 32 bits, e unidade *floating-point*, possuindo este uma frequência máxima de relógio de 667 MHz [15]. Para efetuar alterações neste *software* foi utilizado o SDK da Xilinx.

### 3.3.1. FreeRTOS

O sistema operativo em tempo real incorporado no *software* do microprocessador deste sistema é o FreeRTOS [16], desenvolvido pela Real Time Engineers, Ltd. Com a utilização deste sistema operativo, é possível fazer uma gestão mais eficaz do funcionamento da plataforma e dos seus recursos, garantindo, devido ao seu escalonador, as restrições temporais das várias tarefas do sistema. O funcionamento e escalonamento destas tarefas é descrito na subsecção 3.2.3 relativa ao funcionamento do sistema.

### 3.3.2. lwIP

A gestão da comunicação TCP/IP entre a plataforma e o cliente, seja ele a interface gráfica, um script MATLAB ou um terminal de comunicação, por ex., o Hercules da HW-group.com [17], é efetuada pela pilha de comunicação lwIP [18], desenvolvida por Adam Dunkels. Esta implementação destaca-se por ser bastante leve, ocupando apenas algumas dezenas de *kilobytes* de memória RAM e 40 kB de memória ROM. Visto que não são necessários para esta aplicação os serviços e *overhead* normalmente empregues na comunicação TCP/IP, esta implementação adequa-se para implementar os serviços básicos para a comunicação.

### 3.3.3. Funcionamento do sistema

De forma a garantir os requisitos de funcionamento desta plataforma, o código de execução implementado no microprocessador deve estabelecer a comunicação TCP/IP com o cliente e controlar o fluxo de dados e configurações entre este e a plataforma.

Na Figura 3-3, pode ser visualizado o fluxograma da tarefa Main implementada previamente a esta dissertação.

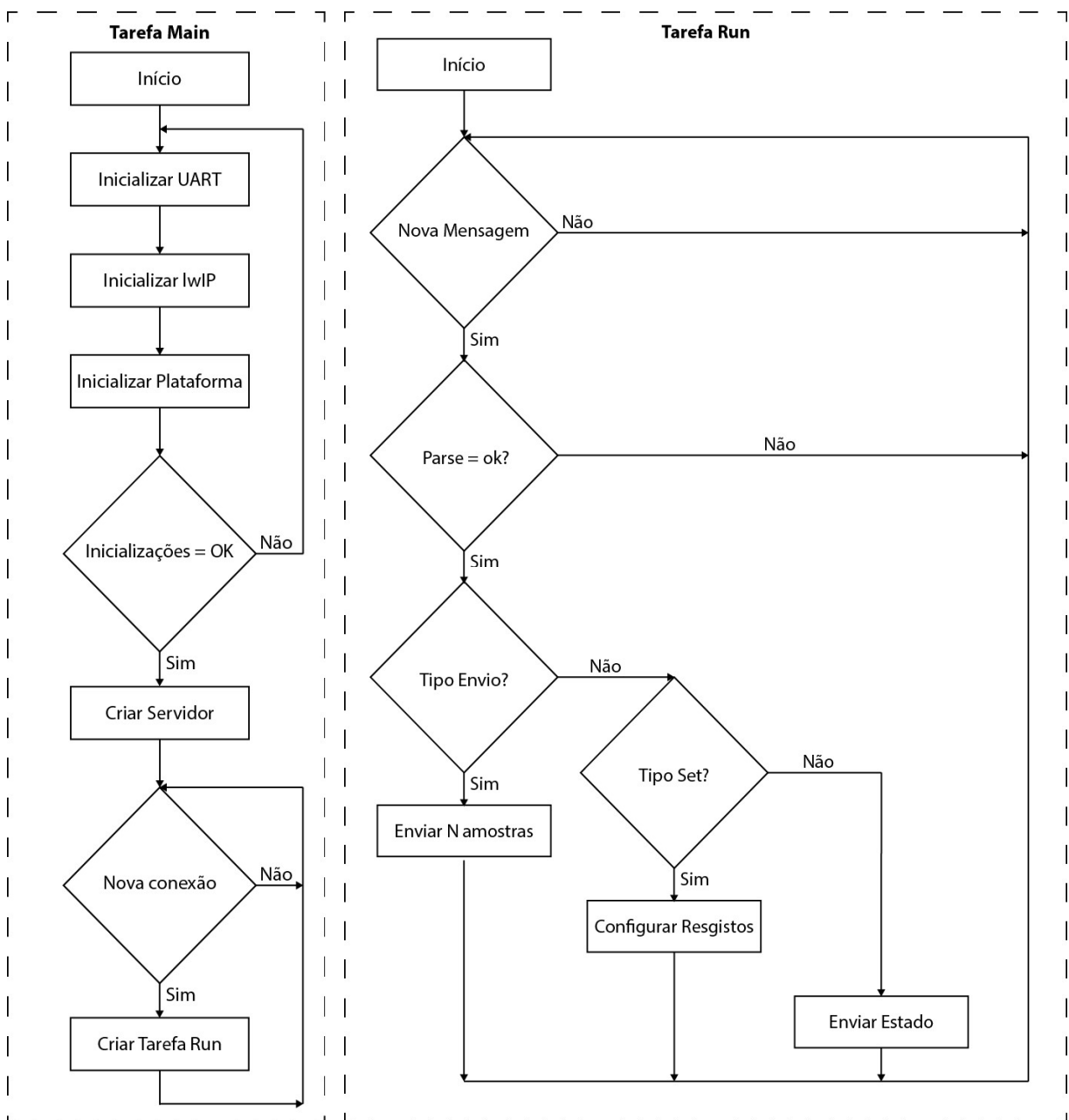


Figura 3-3 - Fluxogramas das tarefas Main e Run previamente implementadas [2].

Como é possível verificar, esta tarefa Main é responsável pela inicialização da plataforma e pela criação do servidor de regência de novos pedidos de conexão. Este servidor cria a tarefa Run, responsável por interpretar e responder aos pedidos recebidos por TCP/IP. Esta aguarda a chegada de uma nova mensagem, decifra-a (*parsing*) e, caso a sintaxe da mensagem esteja correta, executa o comando pedido. Como pode ser concluído pela observação do fluxograma da Figura 3-3. após a conclusão do *parsing*, o algoritmo compara a mensagem recebida com os três tipos de mensagens essenciais, sendo estas:

- Tipo Envio - pedido de um determinado número de amostras;
- Tipo *Set* – pedido de configuração dos parâmetros da plataforma;
- Tipo *Get* – requisição do valor atual do parâmetro da plataforma.

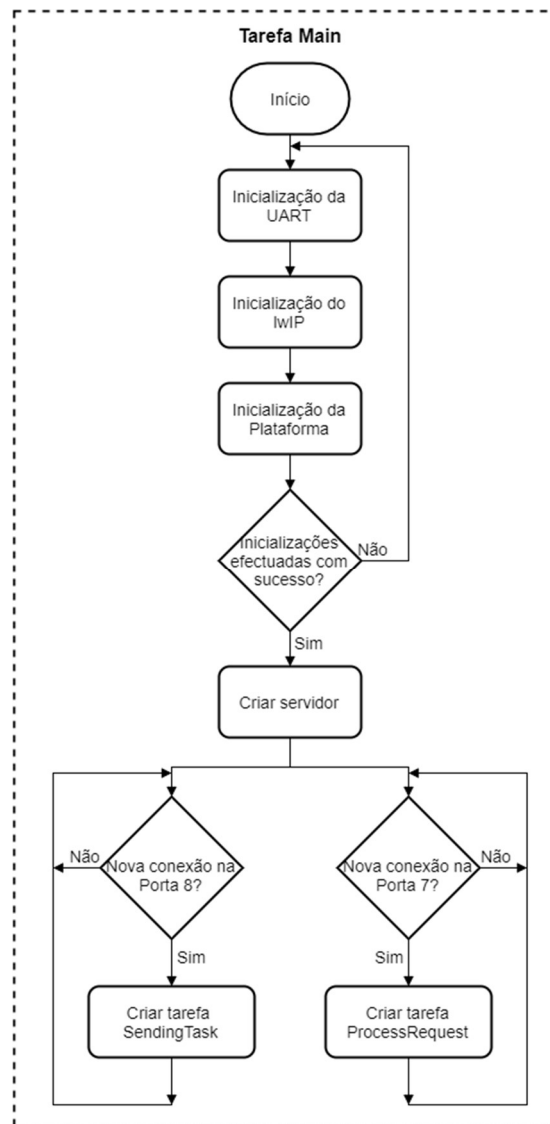


Figura 3-4 - Fluxograma da tarefa Main desenvolvida.

O desenvolvimento desta *toolbox* implicou algumas alterações ao funcionamento das tarefas implementadas. Como pode ser visualizado na Figura 3-4, a reimplementação da tarefa Main permite agora, recorrendo ao uso de sockets, a conexão a duas portas distintas do TCP/IP, sendo estas, a porta 7, correspondente ao echo protocol e cuja conexão foi previamente implementada na plataforma, e a porta 8 de uso genérico. A utilização de sockets permite, através de *file descriptors*, uma configuração e gestão da comunicação mais completas, oferecendo funcionalidades como `read()`, `write()` e `close()` [19].

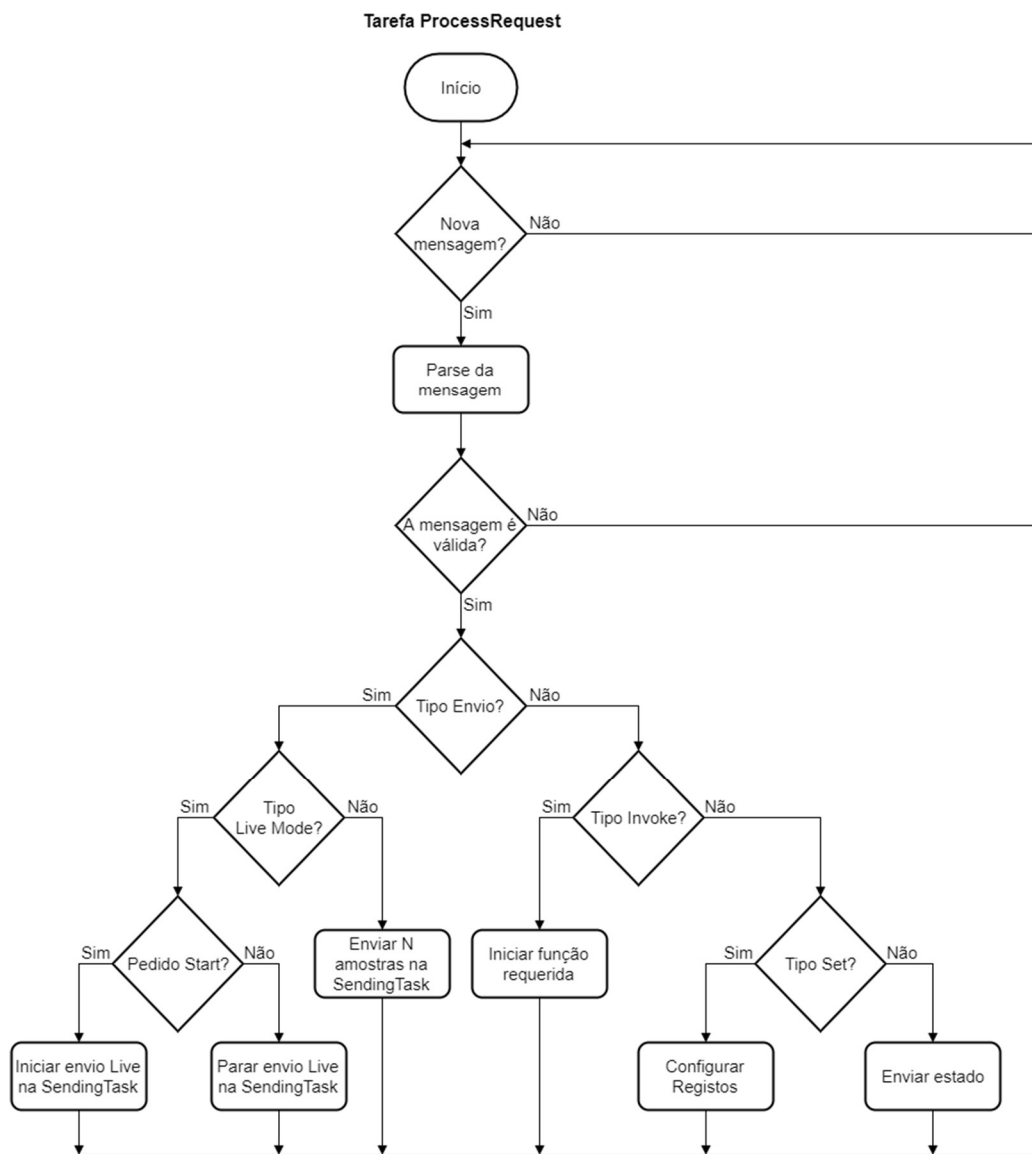


Figura 3-5 - Fluxograma da tarefa ProcessRequest desenvolvida.



A tarefa Run, apresentada na Figura 3-5, agora apelidada ProcessRequest, é conectada à porta 7, tendo as suas funcionalidades sido expandidas, incluindo agora, para além das mencionadas acima, as seguintes mensagens:

- Tipo *invoke* – pedido de execução de funções específicas desenvolvidas neste *software*;
- Tipo *Live Mode* – pedido de início de envio de dados em tempo real e respetivo pedido de término.

A segunda tarefa, denominada SendingTask, é conectada à porta 8 e tem como finalidade o envio dos dados para o cliente. Com a criação desta segunda tarefa, apresentada na Figura 3-6, é possível enviar dados para o cliente e ao mesmo tempo configurar a plataforma. Este paralelismo é especialmente útil para o *Live Mode*, permitindo ao utilizador fazer alterações nos parâmetros da plataforma e visualizar em tempo real as alterações através dos dados adquiridos.

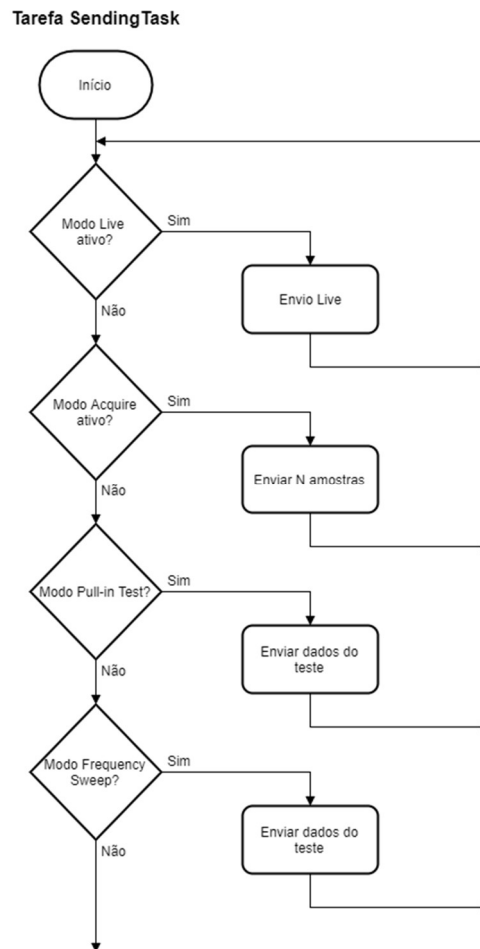


Figura 3-6 - Fluxograma da tarefa SendingTask desenvolvida.

### 3.3.4. Mapeamento dos Registos AXI

Com a finalidade de controlar o *hardware* programável da plataforma, esta contém um banco de registos de configuração mapeado na memória do ARM, podendo este ser acedido através de um valor de *offset* em relação à localização do banco na memória. O valor base dos registos mapeados em memória é o endereço de base 0x43C00000 (XPAR\_RC\_AXI\_INTERFACE\_0\_S00\_AXI\_BASEADDR) sendo o valor de *offset* somado a este de forma a seleccionar o registo desejado.

O protocolo utilizado para aceder a estes registos é o AMBA AXI4 da ARM, mais especificamente a versão AXI4-Lite. Esta versão tem como principais características o facto de ser uma interface de transações singulares, mapeada em memória bastante leve, com uma pegada lógica pequena na FPGA e de uso e implementação simples [14]

Nas Figura 3-7 e Figura 3-8, podem ser analisados os registos de escrita e de leitura da plataforma de teste de MEMS, respetivamente.

Nº de Registo	Registos de Escrita																																Descrição
	bits																																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	Frequência dos dois sinais																																Bloco gerador da portadora
1	Fase entre os dois sinais																																Bloco gerador da portadora
2	Frequência de amostragem																																Bloco ADC Driver
3	Tensão de comparação para o Lado Direito																																Bloco de medição de tempos de Pull-In
4	Tensão de comparação para o Lado Esquerdo																																Bloco de medição de tempos de Pull-In
5	Endereço da memória																Valor para o DAC																Bloco gerador de ondas 0
6	Endereço Inicial																Endereço Final																Bloco gerador de ondas 0
7	Endereço da memória																Valor para o DAC																Bloco gerador de ondas 1
8	Endereço Inicial																Endereço Final																Bloco gerador de ondas 1
9	Endereço da memória																Valor para o DAC																Bloco gerador de ondas 2
10	Endereço Inicial																Endereço Final																Bloco gerador de ondas 2
11	Endereço da memória																Valor para o DAC																Bloco gerador de ondas 3
12	Endereço Inicial																Endereço Final																Bloco gerador de ondas 3
13	Uso Futuro																																Bloco de Reset
14	Uso Futuro																ACTUADOR1	ACTUADOR0	CTRL	TPI	PORT	DAC	ADC	ZG3	WG2	WG1	WGO	MUX DAC3	MUX DAC2	MUX DAC1	MUX DAC0	Bloco de Controlo de Mux's	
15	Tensão de referência																																Bloco de Controlo
16	Tensão de actuação a "High"																																Bloco de Controlo
17	Tensão de actuação a "Low"																																Bloco de Controlo
18	Uso Futuro																																
19	Uso Futuro																																
20	Uso Futuro																																
21	Uso Futuro																																
22	Uso Futuro																																
23	Uso Futuro																SW1 SW1 SW1 SW1																Interruptores Analógicos
24	Uso Futuro																																
25	Uso Futuro																																
26	Uso Futuro																																
27	Uso Futuro																																
28	Uso Futuro																																
29	Uso Futuro																																
30	Uso Futuro																																
31	Uso Futuro																																

Figura 3-7 - Registos de escrita da plataforma de teste de MEMS.

Nº de Registo	Registos de Leitura																Descrição														
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		15	14	13	12	11	10	9	8	7	6	5	4	3	2
0	Tempo medido para a Esquerda																Bloco de medição de tempos de Pull-In														
1	Tempo medido para a Direita																Bloco de medição de tempos de Pull-In														
2	Valor do ADC em 16 bit's Signed																Bloco ADC Driver (signed)														
3	Uso Futuro																														
4	Valor do ADC em 16 bit's Signed amostra N								Valor do ADC em 16 bit's Signed amostra N-1								Bloco ADC Driver buffer 0														
5	Valor do ADC em 16 bit's Signed amostra N-2								Valor do ADC em 16 bit's Signed amostra N-3								Bloco ADC Driver buffer 1														
6	Valor do ADC em 16 bit's Signed amostra N-4								Valor do ADC em 16 bit's Signed amostra N-5								Bloco ADC Driver buffer 2														
7	Valor do ADC em 16 bit's Signed amostra N-6								Valor do ADC em 16 bit's Signed amostra N-7								Bloco ADC Driver buffer 3														
8	Valor do ADC em 16 bit's Signed amostra N-8								Valor do ADC em 16 bit's Signed amostra N-9								Bloco ADC Driver buffer 4														
9	Valor do ADC em 16 bit's Signed amostra N-10								Valor do ADC em 16 bit's Signed amostra N-11								Bloco ADC Driver buffer 5														
10	Valor do ADC em 16 bit's Signed amostra N-12								Valor do ADC em 16 bit's Signed amostra N-13								Bloco ADC Driver buffer 6														
11	Valor do ADC em 16 bit's Signed amostra N-14								Valor do ADC em 16 bit's Signed amostra N-15								Bloco ADC Driver buffer 7														
12	Valor do ADC em 16 bit's Signed amostra N-16								Valor do ADC em 16 bit's Signed amostra N-17								Bloco ADC Driver buffer 8														
13	Valor do ADC em 16 bit's Signed amostra N-18								Valor do ADC em 16 bit's Signed amostra N-19								Bloco ADC Driver buffer 9														
14	Valor do ADC em 16 bit's Signed amostra N-20								Valor do ADC em 16 bit's Signed amostra N-21								Bloco ADC Driver buffer 10														
15	Valor do ADC em 16 bit's Signed amostra N-22								Valor do ADC em 16 bit's Signed amostra N-23								Bloco ADC Driver buffer 11														
16	Valor do ADC em 16 bit's Signed amostra N-24								Valor do ADC em 16 bit's Signed amostra N-25								Bloco ADC Driver buffer 12														
17	Valor do ADC em 16 bit's Signed amostra N-26								Valor do ADC em 16 bit's Signed amostra N-2								Bloco ADC Driver buffer 13														
18	Uso Futuro																														
19	Uso Futuro																														
20	Uso Futuro																														
21	Uso Futuro																														
22	Uso Futuro																														
23	Uso Futuro																														
24	Uso Futuro																														
25	Uso Futuro																														
26	Uso Futuro																														
27	Uso Futuro																														
28	Uso Futuro																														
29	Data _Values																Valor instantâneo da saída do Lock-in														
30	Uso Futuro																														
31	Uso Futuro																														

Figura 3-8 - Registos de leitura da plataforma de teste de MEMS

### 3.3.5. Protocolo de Comunicação

Para que o utilizador possa comunicar com a plataforma através da comunicação TCP/IP, e assim configurar os parâmetros da mesma, é necessário um protocolo de comunicação. Este, para além de funcional, deve ser simples e intuitivo, permitindo ao utilizador da plataforma aceder a esta mesmo sem a necessidade de uma interface gráfica, bastando enviar a mensagem correta diretamente para o IP da plataforma e através da porta desejada (este método direto de comunicação é aprofundado no capítulo 4). De forma a garantir estes requisitos, a estrutura do protocolo de comunicação criado é baseada nos protocolos profissionais GPIB, utilizando uma nomenclatura e estrutura axiomáticas. Por exemplo, para selecionar entre o modo de acoplamento AC e DC do canal de leitura da plataforma, o comando a enviar é 'RC:ACDC:AC', para o caso da seleção do modo AC. A tabela completa das mensagens utilizadas neste protocolo de comunicação pode ser consultada no Anexo A deste documento.

### 3.4. Driver de comunicação

Para garantir que a comunicação entre o utilizador e a plataforma é feita de forma fiável e intuitiva, optou-se por implementar um *driver* de comunicação para efetuar a tradução das mensagens enviadas pelo utilizador para a estrutura do protocolo de comunicação abordado anteriormente na subsecção 3.3.5, sendo o MATLAB Instrument Driver a ferramenta selecionada. Este instrumento faz parte do *software* Instrument Control Toolbox e permite comunicar com um instrumento de *hardware* através de várias interfaces diferentes, sendo a escolhida nesta dissertação, de forma a estar em conformidade com os protocolos de comunicação já implementados, o TCP/IP. A implementação do MATLAB Instrument Driver oferece uma camada intermédia de interpretação entre o utilizador e o *hardware*, neste caso, a plataforma, permitindo que esta seja configurada com comandos mais intuitivos e tornando mais simples redigir, por exemplo, um script MATLAB com um bloco de comandos. Os métodos utilizados para configuração da plataforma com o uso do *driver* de comunicação são descritos no capítulo 4, especificando a sua sintaxe para cada configuração ou chamada de função. A Figura 3-9 demonstra o funcionamento desta ferramenta.

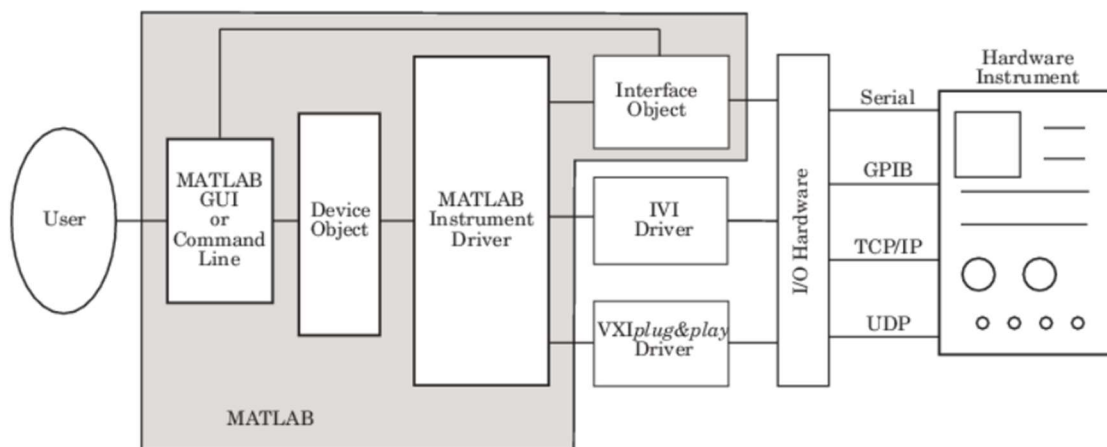


Figura 3-9 - Visão geral do funcionamento do MATLAB Instrument Driver [20].

## 3.5. Interface Gráfica

Para o desenvolvimento da interface gráfica, preferiu-se a utilização de uma ferramenta que fosse totalmente compatível com o MATLAB, de modo a que pudessem ser utilizadas as inúmeras ferramentas de tratamento de dados e comunicação do MATLAB, por ex., Instrument Control Toolbox e Signal Processing Toolbox. O MATLAB disponibiliza dois ambientes de desenvolvimento de interfaces gráficas, sendo estes o GUIDE e o App Designer [21]. Como o App Designer consiste no mecanismo mais recente, apresenta ainda alguma falta de suporte e funcionalidades, sendo este problema compensado pelas seguintes características:

- Maior capacidade de customização da GUI a desenvolver;
- Utilização da estrutura de classes do MATLAB, o que facilita a leitura e escrita do código;
- Editor integrado com inclusão de um *debugger*;
- Utilização de apenas um ficheiro do tipo MLAPP para edição e execução da interface gráfica desenvolvida, não necessitando assim de um ficheiro FIG como é o caso do GUIDE.

Após consideração das vantagens e desvantagens de ambos os ambientes de desenvolvimento, optou-se pelo App Designer para o desenvolvimento desta interface gráfica.

## 3.6. Conclusão do capítulo

Neste capítulo, foi feita uma análise às componentes que constituem a plataforma de teste de MEMS e alterações efetuadas à arquitetura desta no âmbito desta dissertação, assim como uma breve apreciação das ferramentas a utilizar no seu desenvolvimento.



# 4.

## Funcionalidades da *Toolbox*

Neste capítulo, são descritas as funcionalidades da *toolbox*, com especial ênfase na explicação do seu funcionamento, modo de utilização e especificações da mesma.

A primeira secção faz uma descrição global da *toolbox* e dos métodos utilizados, incluindo fluxogramas genéricos destes. As restantes secções são divididas de acordo com os painéis da interface gráfica da *toolbox*, incorporando, para cada uma dessas configurações, tabelas com informação detalhada da mesma, que juntamente com o correspondente fluxograma descrito na secção 4.1, explicam o funcionamento de cada painel.

## 4.1. Visão geral do sistema

Depois de selecionadas as ferramentas a utilizar com o modelo de funcionamento definido, iniciou-se o desenvolvimento da interface gráfica (GUI), do *driver* de comunicação e das restantes adaptações necessários no *software* do microprocessador.

A interface gráfica, implementada utilizando o App Designer [22], foi criada visando obter a máxima funcionalidade e facilidade de utilização. O painel principal desta GUI pode ser visualizado na Figura 4-1, onde estão representadas as três principais secções desta: no topo, o painel de configuração da comunicação e o painel de controlo dos interruptores TTL; na secção intermédia, os painéis de funções de aquisição; e na parte inferior, os painéis de configuração. Estes painéis são descritos nas secções que se seguem.

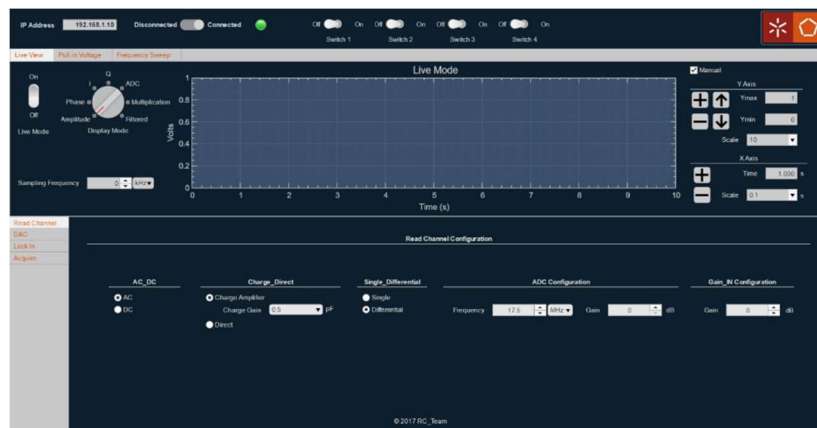


Figura 4-1 - Painel principal da interface gráfica.

De forma a munir a interface gráfica com a capacidade de configurar a plataforma de teste de MEMS, foram desenvolvidos três modelos funcionais, dois de configuração de parâmetros, *set* e *get*, e outro de invocação de funções, *invoke*. Para introduzir estes métodos, explicando como utilizá-los e alterá-los, estes serão descritos neste documento através de fluxogramas genéricos. Nas secções seguintes, relativas à explicação do funcionamento de cada um dos painéis, foram adicionadas tabelas com os números correspondentes, podendo estes ser usados para adaptar o fluxograma a cada painel.



Uma das vantagens da implementação desta estrutura na *toolbox* é a possibilidade de utilizar os métodos desenvolvidos, a partir da entrada de qualquer um dos blocos, ou seja, utilizando o método *set* como exemplo, este pode ser invocado das seguintes formas:

- Pela interação com o painel da interface gráfica;
- Através de um *script* MATLAB com o método de chamada (`set(H,'PropertyName','PropertyValue')`);
- Diretamente pelo envio da mensagem correspondente do protocolo de comunicação, utilizando por exemplo, um terminal de comunicação TCP/IP ligado à porta 7.

### *Método Set*

A Figura 4-2 demonstra o funcionamento do método *set*. De forma a complementá-lo, este será descrito com recurso a um exemplo. Novamente para o caso do modo de acoplagem, o utilizador inicia o processo efetuando a seleção do valor pretendido no painel AC\_DC, escolhendo por exemplo, o modo AC. Esta seleção é validada na interface gráfica o que dá início ao *callback* deste painel, procedendo este à chamada do método *set* (`set(App.deviceObj.RC1,'AC_DC','AC')`). Este pedido é enviado para o *driver* que procede a uma nova validação da seleção pretendida. Sendo esta válida, o valor é convertido num valor aceite pelo instrumento, seguindo-se o envio de uma mensagem pela porta 7 com o pedido *get* ('RC1:ACDC?'), de forma a verificar se o valor pretendido já está definido. No microprocessador é feito o *parsing* desta mensagem e a leitura deste valor utilizando o método *get* correspondente. Este valor é enviado por mensagem ('DC') pela porta 7 para o *driver*. Se a opção definida for diferente do pedido, como é o caso deste exemplo, é iniciado o envio da mensagem *set* ('RC1:ACDC:AC') pela porta 7, procedendo-se o *parsing* deste valor. O método *set* adequado é selecionado e o valor é novamente validado, sendo a configuração concluída se este for válido.

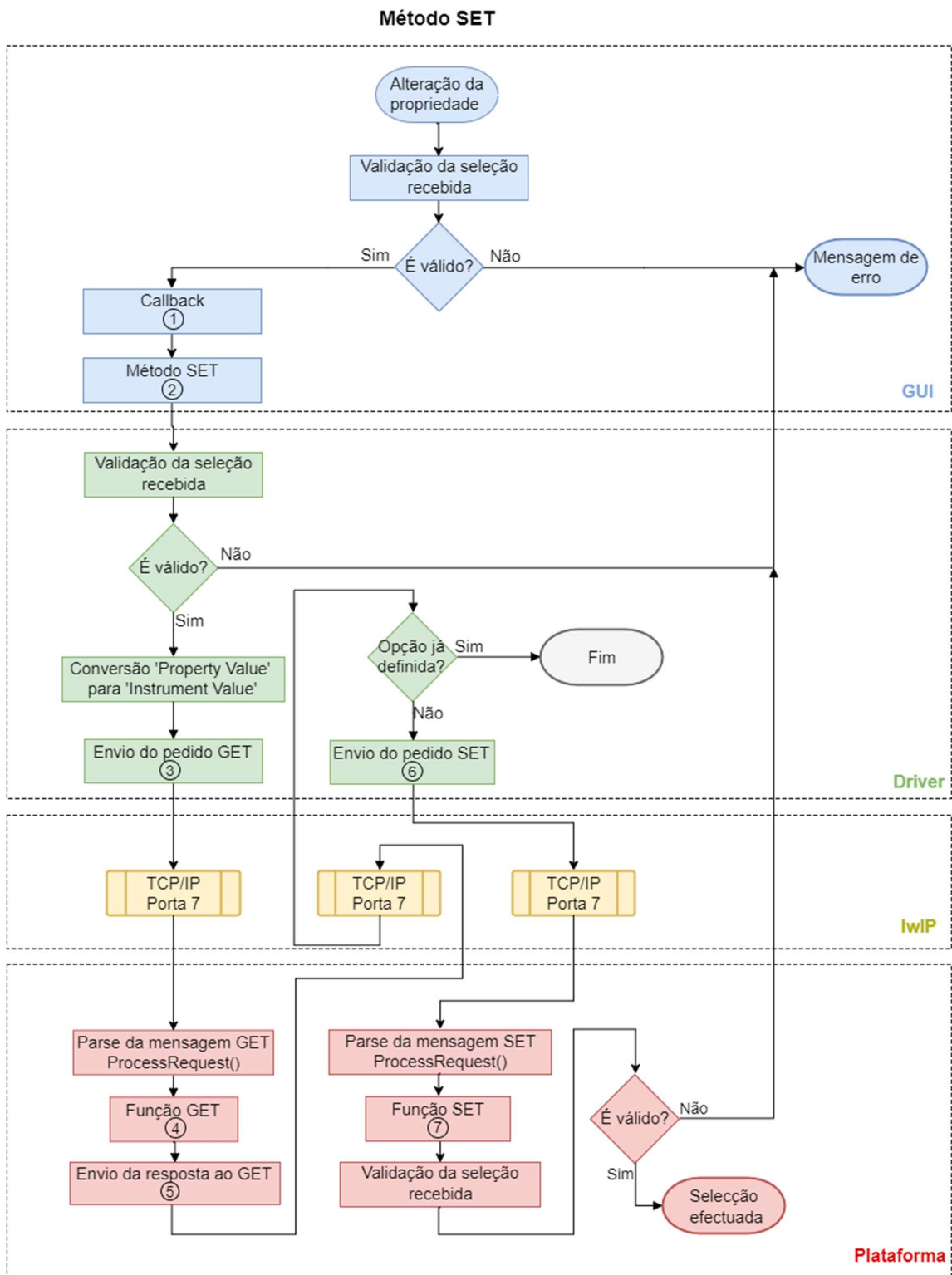


Figura 4-2 - Fluxograma do método set.

### Método Get

Na Figura 4-3, pode ser visualizado o fluxograma relativo ao método *get*. O funcionamento deste método será descrito de seguida com o auxílio de um exemplo. Para o caso do modo de acoplagem, o primeiro passo é a chamada do método *get* (`value=get(App.deviceObj.RC1,'AC_DC')`). O *driver* inicia assim a transmissão da mensagem definida no protocolo de comunicação ('RC1:ACDC?') e envia-a pela porta 7 da comunicação TCP/IP. A mensagem é recebida no microprocessador e o seu *parsing* é efetuado na tarefa *ProcessRequest*. Depois de identificado o pedido no *parser*, é efetuada a leitura do valor desta propriedade e enviada a resposta (ex. 'AC') novamente pela porta 7. O *driver* recebe a resposta e converte este valor do instrumento (*InstrumentValue*) no valor de saída da propriedade (*PropertyValue*), atribuindo-o à variável de retorno do método *get*.

Método GET

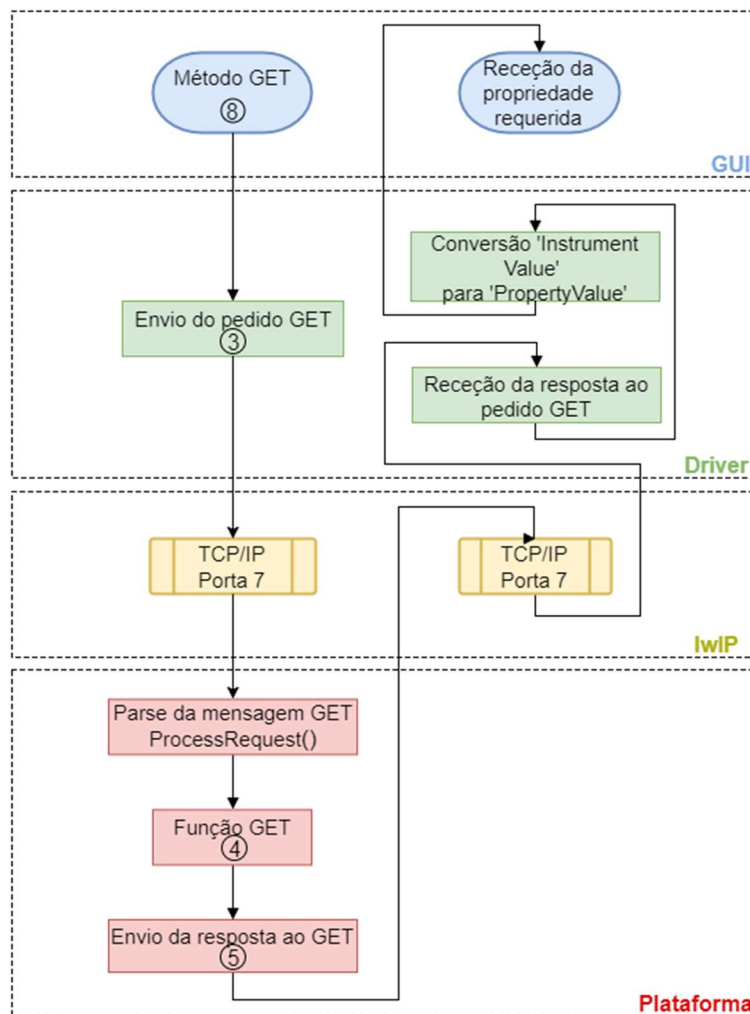


Figura 4-3 - Fluxograma do método *get*.

### Método *Invoke*

O fluxograma do método *invoke* é apresentado na Figura 4-4. A análise do seu funcionamento será feita recorrendo a um exemplo. De forma a pedir 500 amostras da amplitude de um sinal, o utilizador seleciona o número de amostras a pedir e o modo de aquisição e pressiona o botão de início na interface gráfica. Esta ação desbloqueia a chamada do *callback* respetivo, chamando este o método *invoke* (`invoke('App.deviceObj.ACQ1','Amplitude',app.Samples)`). Este método ativa o *driver* que procede ao envio da mensagem correspondente a este pedido pela porta 7 do TCP/IP ('GET:AMP:10000'). No microprocessador, é efetuado o *parsing* desta mensagem e os parâmetros do pedido são validados, neste caso, o número de amostra a receber. Se este for válido, é iniciado o envio das amostras pela *SendingTask*, através da porta 8. Estes dados são então recebidos na interface gráfica, não passando pelo *driver*, dado que estes são enviados pela porta de dados. Depois de recebidas, as amostras são apresentadas num gráfico na interface gráfica.

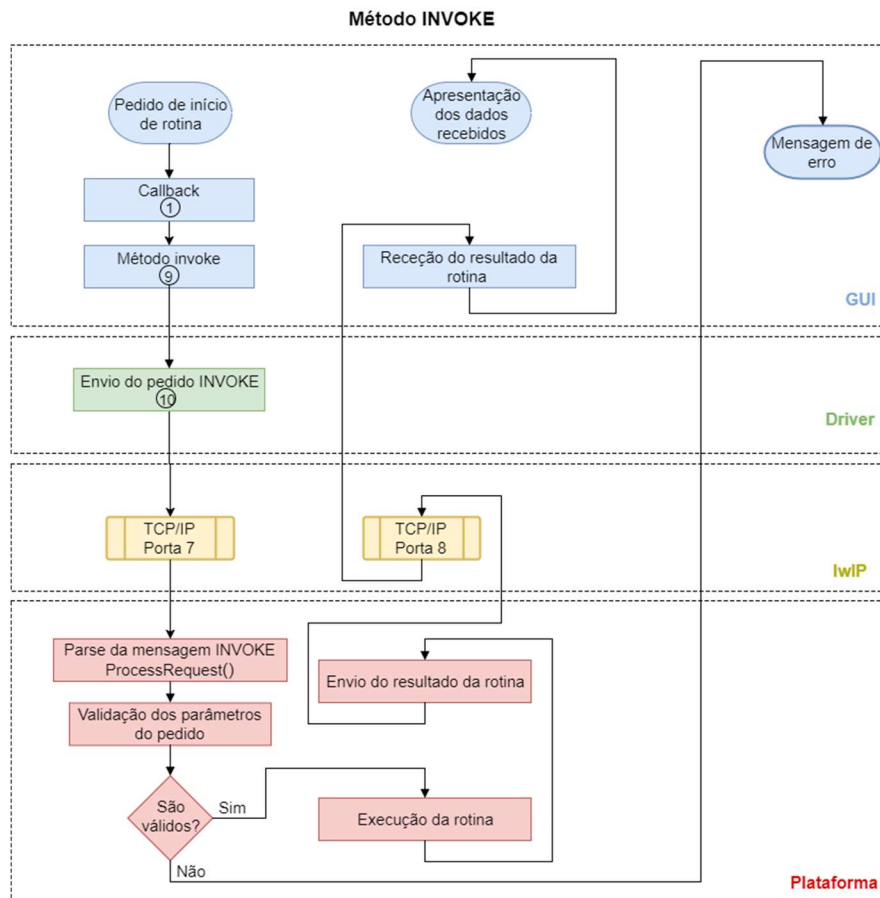


Figura 4-4 - Fluxograma do método *invoke*.

## 4.2. Painel Read Channel

De modo a possibilitar a configuração das propriedades do sistema de aquisição, foi necessário incorporar na interface gráfica um painel para este efeito, juntamente com a implementação dos respetivos métodos no *driver* e microprocessador.



Figura 4-5 - Painel Read Channel Configuration.

Este sistema de leitura é composto por um amplificador de carga, um conversor diferencial, um amplificador de sinal e um conversor analógico-digital, podendo a sua arquitetura ser conferida na Figura 4-6.

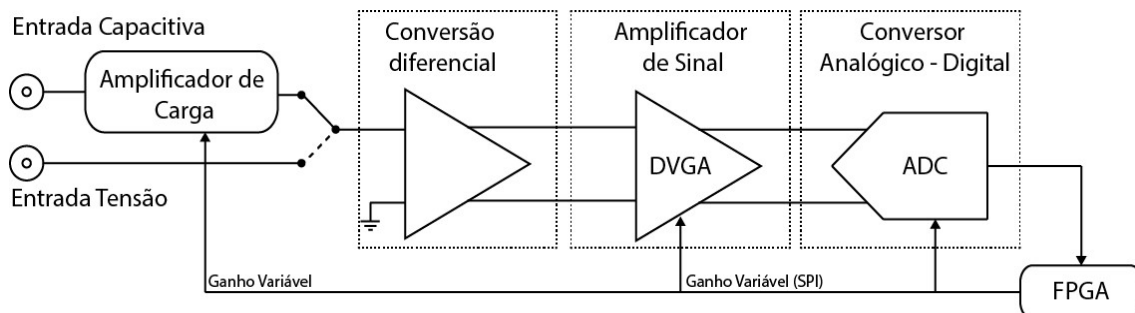


Figura 4-6 - Arquitetura do sistema de aquisição da plataforma de teste de MEMS[2].

Como pode ser observado na Figura 4-5, este painel encontra-se dividido em vários blocos de customização, nomeadamente:

- Acoplamento AC/DC;
- Seleção de entrada em tensão ou capacitiva (amplificador de carga);
- Seleção *single ended* ou diferencial;
- Configuração do ADC;
- Seleção do ganho de entrada.

Estes blocos são descritos com mais detalhe nas subsecções seguintes, minuciando a sua finalidade e funcionamento.

### 4.2.1. Configuração AC/DC

Uma das características deste sistema de aquisição é permitir a seleção entre acoplamento AC e DC, sendo necessária uma interface para alternar entre estes dois modos. Para esse efeito, foi criado, na interface gráfica, um painel de configuração denominado “AC\_DC”. Este painel, que pode ser visualizado na Figura 4-7, apresenta duas opções de seleção, AC e DC.

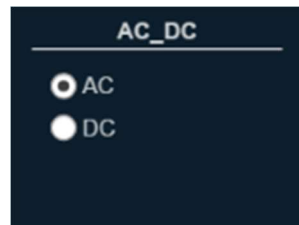


Figura 4-7 - Painel AC\_DC do separador Read Channel.

Utilizando o modelo de configuração apresentado na secção 4.1 em conjunto com os métodos descritos na Tabela 4-1, foi possível proceder à configuração do modo de acoplamento, garantindo a correta configuração do mesmo, através da interface gráfica, por um *script* MATLAB, utilizando o método *set* ou diretamente, por TCP/IP, utilizando a porta 7 e a mensagem de comunicação listada na Tabela 4-2.

Tabela 4-1 - Métodos AC\_DC.

GUI		
Callback	RC(i)_AC_DC_SelectionChanged () i: índice do RC	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'AC_DC' PropertyValue: Tabela 4-2	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'AC_DC' value: Tabela 4-2	⑧
Driver		
Set	'RC(i):ACDC:(Instrument Value)##' i: índice do RC Instrument Value: Tabela 4-2 ex: 'RC1:ACDC:AC##'	⑥
Get	'RC(i):ACDC?##' i: índice do RC ex: 'RC1:ACDC?##'	③
Plataforma		
Set	Set_IN_ACDC(READ_CHANEL *d, float Value) READ_CHANEL *d: Apontador para a estrutura do canal de leitura float Value: Tabela 4-2	⑦
Get	float Value =Get_IN_ACDC(READ_CHANEL *d) READ_CHANEL *d: Apontador para a estrutura do canal de leitura	④
	'(Instrument_Value)##' Instrument_Value: Tabela 4-2	⑤

Na configuração do modo de acoplamento, os valores dos métodos *set* (Property Value) e os valores a enviar na mensagem de comunicação (Instrument Values) são do tipo *string* e estão listados na Tabela 4-2, juntamente com o valor correspondente do tipo *float*, no microprocessador.

Tabela 4-2 - Valores permitidos para a configuração do AC\_DC.

Property Value	Instrument Value	Value
AC	AC	1
DC	DC	0

## 4.2.2. Configuração Amplificador de Carga

Para tornar possível a transdução de capacidades em tensões, e assim obter leituras de um sensor ou atuador, a plataforma de teste de MEMS inclui, no sistema de leitura, um bloco amplificador de carga. Este bloco utiliza uma tensão alternada de excitação nos terminais do condensador em conjunto com um amplificador de transimpedância para converter a variação da corrente numa variação de tensão.

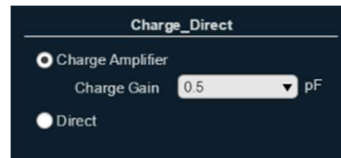


Figura 4-8 - Painel Charge\_Direct do separador Read Channel.

Com o intuito de permitir a seleção entre o amplificador de carga (entrada capacitiva) e o modo direto (entrada em tensão), foi criado um painel na interface gráfica, denominado “Charge\_Direct” e que pode ser visualizado na Figura 4-8.

Tabela 4-3 - Métodos Charge\_Direct.

GUI		
Callback	RC(i)_Charge_Direct_SelectionChanged () i: índice do RC	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'CHARGE_DIRECT' PropertyValue: Tabela 4-4	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'CHARGE_DIRECT' value: Tabela 4-4	⑧
Driver		
Set	'RC(i):CD:(Instrument Value)##' i: índice do RC Instrument Value: Tabela 4-4 ex: 'RC1:CD:C##'	⑥
Get	'RC(i):CD?##' i: índice do RC ex: 'RC1:CD?##'	③
Plataforma		
Set	Set_IN_CHG_DIR(READ_CHANEL *d, float Value) READ_CHANEL *d: Apontador para a estrutura do canal de leitura float Value: Tabela 4-4	⑦
Get	float Value =Get_IN_CHG_DIR(READ_CHANEL *d) READ_CHANEL *d: Apontador para a estrutura do canal de leitura	④
	'(Instrument_Value)##' Instrument_Value: Tabela 4-4	⑤



A Tabela 4-3 e a Tabela 4-4 explicitam os métodos utilizados e os valores aceites na seleção do modo de aquisição, respetivamente.

Tabela 4-4 - Valores permitidos para a configuração do Charge\_Direct.

Property Value	Instrument Value	Value
Charge Amplifier	C	0
Direct	D	1

Sendo este um amplificador de carga de ganho variável, possui também um conjunto de relés que permitem variar a capacidade da malha de realimentação, variando assim o ganho do sistema. Para este efeito, a configuração do amplificador de carga permite também a alteração do ganho a utilizar. Os valores permitidos são enumerados abaixo, na Tabela 4-6, juntamente com as funções e mensagens utilizadas. Ambas as configurações seguem o modelo de configuração detalhado na secção 4.1 e os métodos da Tabela 4-5.

Tabela 4-5 - Métodos Charge\_Gain.

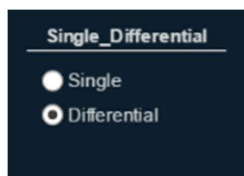
GUI		
Callback	RC(i)_Charge_Gain_SelectionChanged () i: índice do RC	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'CHARGE_GAIN' PropertyValue: Tabela 4-6	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'CHARGE_GAIN' value: Tabela 4-6	⑧
Driver		
Set	'RC(i):CG:(Instrument Value)##' i: índice do RC Instrument Value: Tabela 4-6 ex: 'RC1:CG:0.5##'	⑥
Get	'RC(i):CG?##' i: índice do RC ex: 'RC1:CG?##'	③
Plataforma		
Set	Set_IN_CHGGain(READ_CHANEL *d, float Value) READ_CHANEL *d: Apontador para a estrutura do canal de leitura float Value: Tabela 4-6	⑦
Get	float Value =Get_IN_CHGGain(READ_CHANEL *d) READ_CHANEL *d: Apontador para a estrutura do canal de leitura	④
	'(Instrument_Value)##' Instrument_Value: Tabela 4-6	⑤

Tabela 4-6 - Valores permitidos para a configuração do *Charge\_Gain*.

Property Value	Instrument Value	Value
0.5	14	14
1.0	13	13
1.5	12	12
2.0	11	11
2.5	10	10
3.0	9	9
3.5	8	8
3.9	7	7
4.4	6	6
4.9	5	5
5.4	4	4
5.9	3	3
6.4	2	2
6.9	1	1
7.4	0	0

### 4.2.3. Configuração *Single Ended*/Diferencial

Como pode ser visualizado na Figura 4-6, posteriormente às entradas em tensão e capacidade/corrente, a plataforma possui um amplificador diferencial. Este amplificador tem o propósito de converter as entradas *single ended* para diferencial, para que se tornem compatíveis com os restantes subsistemas (ADC e DVGA) e garantindo uma maior imunidade ao ruído. Para possibilitar a escolha entre os modos *single ended* e diferencial, desenvolveu-se um mecanismo de configuração semelhante ao utilizado na seleção do modo de acoplamento. O painel “*Single\_Differential*” pode ser observado na Figura 4-9.

Figura 4-9 - Painel *Single\_Differential* do separador *Read Channel*.

A Tabela 4-7 e a Tabela 4-8 especificam os métodos *set* e *get* utilizados na interface gráfica, as mensagens de comunicação e as funções *set* e *get* no microprocessador que permitem a configuração desta propriedade, assim como os valores permitidos, respetivamente.

Tabela 4-7 - Métodos Single\_Differential.

GUI		
Callback	RC(i)_S_D_SelectionChanged () i: índice do RC	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'SINGLE_DIFFERENTIAL' PropertyValue: Tabela 4-8	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'SINGLE_DIFFERENTIAL' value: Tabela 4-8	⑧
Driver		
Set	'RC(i):SD:(Instrument Value)##' i: índice do RC Instrument Value: Tabela 4-8 ex: 'RC1:SD:S##'	⑥
Get	'RC(i):SD?##' i: índice do RC ex: 'RC1:SD?##'	③
Plataforma		
Set	Set_IN_SNG_DIF(READ_CHANEL *d, float Value) READ_CHANEL *d: Apontador para a estrutura do canal de leitura float Value: Tabela 4-8	⑦
Get	float Value =Get_IN_SNG_DIF(READ_CHANEL *d) READ_CHANEL *d: Apontador para a estrutura do canal de leitura  '(Instrument_Value)##' Instrument_Value: Tabela 4-8	④ ⑤

Tabela 4-8 - Valores permitidos para a configuração do Single\_Differential.

Property Value	Instrument Value	Value
Single	S	1
Differential	D	0

#### 4.2.4. Configuração do ganho de entrada

Para viabilizar a medição de variação de sinais mais pequenos, o sistema de leitura incorpora um amplificador de ganho variável, permitindo aumentar dinamicamente a amplitude do sinal de entrada e assim tentar garantir que a sua variação é superior à resolução do ADC.



Figura 4-10 - Painel Gain\_IN Configuration do separador Read Channel.

O *Digital Variable Gain Amplifier* (DVGA) utilizado permite a variação do ganho entre -6 dB e 26 dB e, como meio de alteração deste valor, foi incorporado nesta *toolbox* um mecanismo de configuração baseado no modelo descrito na secção 4.1, sendo os seus métodos específicos apresentados na Tabela 4-9.

Tabela 4-9 - Métodos Gain\_In.

GUI		
Callback	RC(i)_Gain_ValueChanged () i: índice do RC	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'GAIN' PropertyValue: <i>float</i> entre -6,0 e 26,0 dB	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'GAIN' value: <i>float</i> entre -6,0 e 26,0 dB	⑧
Driver		
Set	'RC(i):GAIN:(Instrument Value)##' i: índice do RC Instrument Value: <i>float</i> entre -6,0 e 26,0 dB ex: 'RC1:GAIN:-6.0##'	⑥
Get	'RC(i):GAIN?##' i: índice do RC ex: 'RC1:GAIN?##'	③
Plataforma		
Set	Set_IN_Gain(READ_CHANEL *d, <i>float</i> Value) READ_CHANEL *d: Apontador para a estrutura do canal de leitura <i>float</i> Value: <i>float</i> entre -6,0 e 26,0 dB	⑦
Get	<i>float</i> Value =Get_IN_Gain(READ_CHANEL *d) READ_CHANEL *d: Apontador para a estrutura do canal de leitura  '(Instrument_Value)##' Instrument_Value: <i>float</i> entre -6,0 e 26,0 dB	④ ⑤

O ganho de entrada do canal de leitura pode ser definido utilizando valores do tipo *float* entre -6,0 e 26,0 dB.

### 4.2.5. Configuração do ADC

O último estágio do sistema de aquisição é a conversão dos sinais analógicos lidos para digital. Para este efeito, a plataforma de teste engloba um conversor analógico-digital (ADC), com frequência máxima de amostragem de 40 MHz e 16 bits de resolução.



Figura 4-11 - Painel ADC Configuration do separador Read Channel.

Visando dar ao utilizador da *toolbox* a possibilidade de alterar estas componentes, foi desenvolvido um mecanismo de configuração do funcionamento do ADC. Este processo de customização, explicitado na secção 4.1, utiliza os métodos expostos na Tabela 4-10 e na Tabela 4-12, e permite a modificação das propriedades do ADC. O ganho do ADC pode ser configurado com os valores especificados na Tabela 4-11.

Tabela 4-10 - Métodos ADC\_Gain.

GUI		
Callback	RC(i)_ADC_Gain_ValueChanged () i: índice do RC	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'ADC_GAIN' PropertyValue: Tabela 4-11	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'ADC_GAIN' value: Tabela 4-11	⑧
Driver		
Set	'RC(i):ADC:GAIN:(Instrument Value)##' i: índice do RC Instrument Value: Tabela 4-11 ex: 'RC1:ADC:GAIN:3.0##'	⑥
Get	'RC(i):ADC:GAIN?##' i: índice do RC ex: 'RC1:ADC:GAIN?##'	③
Plataforma		
Set	Set_IN_ADCGain(READ_CHANEL *d, float Value) READ_CHANEL *d: Apontador para a estrutura do canal de leitura float Value: Tabela 4-11	⑦
Get	float Value =Get_IN_ADCGain(READ_CHANEL *d) READ_CHANEL *d: Apontador para a estrutura do canal de leitura	④
	'(Instrument_Value)##' Instrument_Value: Tabela 4-11	⑤

Tabela 4-11 - Valores permitidos para a configuração do ADC\_Gain.

Property Value	Instrument Value	Value
0.0	0.0	0.0
1.0	1.0	1.0
2.0	2.0	2.0
3.0	3.0	3.0
4.0	4.0	4.0
5.0	5.0	5.0
6.0	6.0	6.0

Tabela 4-12 - Métodos ADC\_Freq.

GUI		
Callback	RC(i)_ADC_Freq_ValueChanged () i: índice do RC	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'ADC_FREQ' PropertyValue: <i>float</i> positivos até 17,5 MHz.	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.RC1) 'PropertyName': 'ADC_FREQ' value: <i>float</i> positivos até 17,5 MHz.	⑧
Driver		
Set	'RC(i):ADC:FREQ:(Instrument Value)##' i: índice do RC Instrument Value: <i>float</i> positivos até 17,5 MHz. ex: 'RC1:ADC:FREQ:10000##'	⑥
Get	'RC(i):ADC:FREQ?##' i: índice do RC ex: 'RC1:ADC:FREQ?##'	③
Plataforma		
Set	Set_IN_ADCFreq(READ_CHANEL *d, <i>float</i> Value) READ_CHANEL *d: Apontador para a estrutura do canal de leitura <i>float</i> Value: <i>float</i> positivos até 17,5 MHz.	⑦
Get	<i>float</i> Value =Get_IN_ADCFreq(READ_CHANEL *d) READ_CHANEL *d: Apontador para a estrutura do canal de leitura	④
	'(Instrument_Value)##' Instrument_Value: <i>float</i> positivos até 17,5 MHz.	⑤

A frequência do ADC pode ser definida utilizando valores do tipo *float* positivos até 17,5 MHz.

### 4.3. Painel DAC

Com o objetivo de tornar possível atuar os dispositivos MEMS a testar, a plataforma de teste inclui um sistema de atuação. Este sistema, apresentado na Figura 4-12, é composto por um bloco de quatro interruptores analógicos, quatro DACs e um DDS (*Digital Direct Synthesizer*) para geração das portadoras.

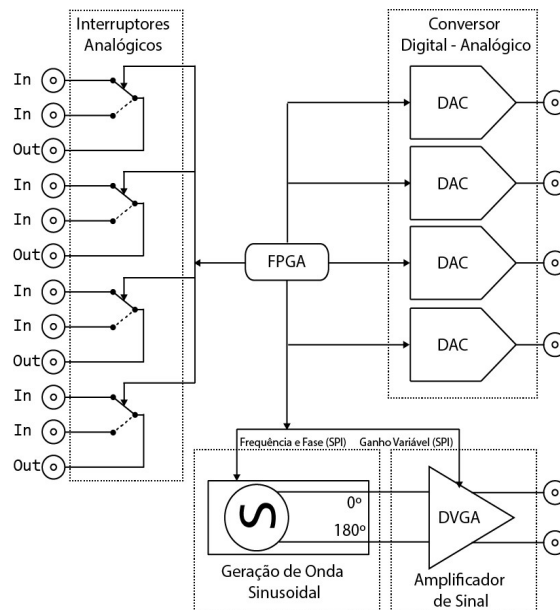


Figura 4-12 - Arquitetura do sistema de atuação da plataforma de teste de MEMS [2]

O painel DAC foi desenvolvido para possibilitar ao utilizador configurar algumas das propriedades deste sistema de atuação, nomeadamente:

- Modo de funcionamento de cada DAC
- Configuração do gerador de ondas
  - Tipo de onda
  - Frequência
  - Amplitude
  - Fase
  - *Offset*
  - *Duty Cycle*
- Interruptores analógicos

Estes blocos de configuração são descritos com detalhe nas subsecções seguintes.

### 4.3.1. Seleção do modo de funcionamento do DAC

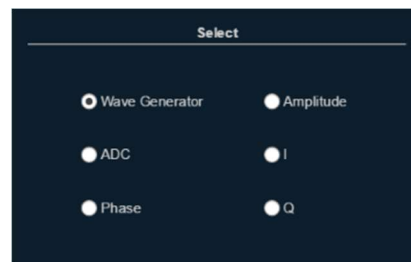


Figura 4-13 Painel Select do separador DAC.

Para possibilitar a seleção do output dos DACs, foi criado o painel ‘Select’ na configuração dos DACs que, juntamente com os métodos desenvolvidos no *driver* e no *software* do microprocessador descritos na secção 4.1, permitem escolher entre as seguintes opções:

- Gerador de onda
- ADC
- Saída em amplitude do *lock-in*
- Saída em fase do *lock-In*
- I (*In-Phase Component*)
- Q (*Quadrature Component*)

Este painel, representado na Figura 4-13, utiliza os métodos explicitados na Tabela 4-13 e os valores da Tabela 4-14.



Tabela 4-13 - Métodos DAC\_Select.

GUI		
Callback	DAC(i)_Sel_SelectionChanged () i: índice do DAC	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'CHARGE_GAIN' PropertyValue: Tabela 4-14	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'SELECT' value: Tabela 4-14	⑧
Driver		
Set	'DAC(i):SEL:(Instrument Value)##' i: índice do DAC Instrument Value: Tabela 4-14 ex: 'DAC1:SEL:WG##'	⑥
Get	'DAC(i):SEL?##' i: índice do DAC ex: 'DAC1:SEL?##'	③
Plataforma		
Set	Set_DAC_SEL(DAC *d, int id, float Value) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC float Value: Tabela 4-14	⑦
Get	float Value =Get_DAC_SEL(DAC *d, int id) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC	④
	'(Instrument_Value)##' Instrument_Value: Tabela 4-14	⑤

Tabela 4-14 - Valores permitidos para a configuração do DAC\_Select.

Property Value	Instrument Value	Value
Wave Generator	WG	3
ADC	ADC	0
Phase	PHS	2
Amplitude	AMP	1
I	I	4
Q	Q	5

### 4.3.2. Configuração do Wave Generator

Para o caso da opção “Wave Generator”, é necessário habilitar o utilizador de uma forma prática de configurar o gerador de ondas. Assim, foram desenvolvidos, na interface gráfica, no *driver* e no *software* do microprocessador, mecanismos para permitir ao utilizador da plataforma escolher a onda desejada a aplicar no DAC. As opções de configuração possíveis são apresentadas em seguida, demonstrando o aspeto de cada uma na GUI, bem como as tabelas de especificação dos métodos criados e os valores de configuração. O painel de configuração do gerador de ondas é apresentado na Figura 4-14.

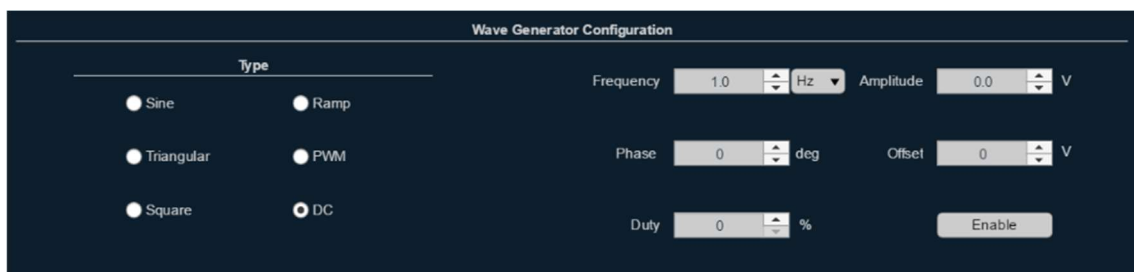


Figura 4-14 - Painel Wave Generator Configuration do separador DAC.

#### *Tipos de onda*

O painel “Type” permite escolher entre seis tipos de ondas, uma DC e cinco AC, nomeadamente, senoide, rampa, triangular, PWM e quadrada.

A Figura 4-15 representa o painel de seleção do tipo de onda gerada.

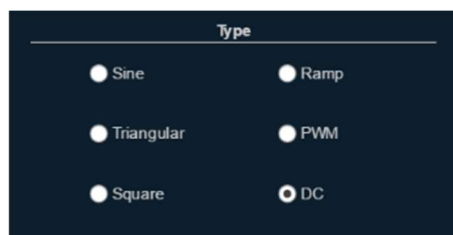


Figura 4-15 - Painel Type do separador Wave Generator.

Na Tabela 4-15 e na Tabela 4-16, são especificados os métodos utilizados na configuração de tipo de onda e os valores aceites em cada uma das fases, respetivamente.

Tabela 4-15 - Métodos WG\_Type.

GUI		
Callback	WG1_Type_SelectionChanged () i: índice do WG	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_TYPE' PropertyValue: Tabela 4-16	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_TYPE' value: Tabela 4-16	⑧
Driver		
Set	'DAC(i):WG:TYPE:(Instrument Value)##' i: índice do DAC Instrument Value: Tabela 4-16 ex: 'DAC1:WG:TYPE:SIN##'	⑥
Get	'DAC(i):WG:TYPE?##' i: índice do DAC ex: 'DAC1:WG:TYPE?##'	③
Plataforma		
Set	Set_DAC_WG_Type(DAC *d, int id, float Value) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC float Value: Tabela 4-16	⑦
Get	float Value =Get_DAC_WG_Type(DAC *d, int id) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC	④
	'(Instrument_Value)##' Instrument_Value: Tabela 4-16	⑤

Tabela 4-16 - Valores permitidos para a configuração do WG\_Type.

Property Value	Instrument Value	Value
Sine	SIN	0
Triangular	TRI	1
Square	SQR	2
Ramp	RAMP	4
PWM	PWM	3
DC	DC	5

### Frequência da onda gerada

A escolha da frequência do sinal a gerar é feita através do painel apresentado na Figura 4-16, utilizando os métodos definidos na Tabela 4-17.

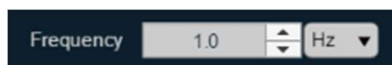


Figura 4-16 - Painel Frequency do separador Wave Generator.

Tabela 4-17 - Métodos WG\_Freq.

GUI		
Callback	WG1_Freq_ValueChanged () i: índice do WG	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_FREQ' PropertyValue: <i>float</i> positivo até 7 MHz.	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_FREQ' value: <i>float</i> positivo até MHz.	⑧
Driver		
Set	'DAC(i):WG:FREQ:(Instrument Value)##' i: índice do DAC Instrument Value: <i>float</i> positivo até 7 MHz. ex: 'DAC1:WG:FREQ:10000##'	⑥
Get	'DAC(i):WG:FREQ?##' i: índice do DAC ex: 'DAC1:WG:FREQ?##'	③
Plataforma		
Set	Set_DAC_WG_Freq(DAC *d, int id, float Value) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC float Value: <i>float</i> positivo até 7 MHz.	⑦
Get	float Value =Get_DAC_WG_Freq(DAC *d, int id) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC	④
	'(Instrument_Value)##' Instrument_Value: <i>float</i> positivo até 7 MHz.	⑤

A frequência do sinal gerado neste bloco pode ser definida utilizando valores do tipo *float* positivos até 7 MHz.

### Amplitude da onda gerada

A amplitude da onda pode ser definida através do painel ilustrado na Figura 4-17, que utiliza os métodos elencados na Tabela 4-18 para concluir a configuração.

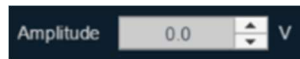


Figura 4-17 - Painel Amplitude do separador Wave Generator.

Tabela 4-18 - Métodos WG\_Amp.

GUI		
Callback	WG1_Amp_ValueChanged () i: índice do WG	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_AMP' PropertyValue: <i>float</i> entre -10 e 10 V	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_AMP' value: <i>float</i> entre -10 e 10 V	⑧
Driver		
Set	'DAC(i):WG:AMP:(Instrument Value)##' i: índice do DAC Instrument Value: <i>float</i> entre -10 e 10 V ex: 'DAC1:WG:AMP:3.4##'	⑥
Get	'DAC(i):WG:AMP?##' i: índice do DAC ex: 'DAC1:WG:AMP?##'	③
Plataforma		
Set	Set_DAC_WG_Amp(DAC *d, int id, float Value) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC <i>float</i> Value: <i>float</i> entre -10 e 10 V	⑦
Get	<i>float</i> Value =Get_DAC_WG_Amp(DAC *d, int id) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC	④
	'(Instrument_Value)##' Instrument_Value: <i>float</i> entre -10 e 10 V	⑤

É possível definir valores de amplitude do tipo *float* entre -10 e 10 V.

### Fase da onda gerada

Para selecionar a fase da onda gerada, o utilizador pode recorrer ao painel de seleção representado na Figura 4-18, sendo os métodos da sua configuração os elencados na Tabela 4-19.



Figura 4-18 - Painel Phase do separador Wave Generator.

Tabela 4-19 - Métodos WG\_Phase.

GUI		
Callback	WG1_Phase_ValueChanged () i: índice do WG	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_PHASE' PropertyValue: <i>float</i> entre -180 e 180	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_PHASE' value: <i>float</i> entre -180 e 180	⑧
Driver		
Set	'DAC(i):WG:PHASE:(Instrument Value)##' i: índice do DAC Instrument Value: <i>float</i> entre -180 e 180 ex: 'DAC1:WG:PHASE:90.0##'	⑥
Get	'DAC(i):WG:PHASE?##' i: índice do DAC ex: 'DAC1:WG:PHASE?##'	③
Plataforma		
Set	Set_DAC_WG_Phase(DAC *d, int id, float Value) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC float Value: <i>float</i> entre -180 e 180	⑦
Get	float Value =Get_DAC_WG_Phase(DAC *d, int id) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC	④
	'(Instrument_Value)##' Instrument_Value: <i>float</i> entre -180 e 180	⑤

A fase do sinal gerado pode ser definida utilizando valores do tipo *float* entre -180 e 180 graus.

### Offset da onda gerada

O utilizador pode definir o valor desejado de *offset da onda gerada*, valendo-se do painel representado na Figura 4-19 ou através de um dos métodos listados na Tabela 4-20.



Figura 4-19 - Painel Offset do separador Wave Generator.

Tabela 4-20 - Métodos WG\_Offset.

GUI		
Callback	WG1_Offset_ValueChanged () i: índice do WG	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_OFFSET' PropertyValue: <i>float</i> entre -10 e 10 V	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_OFFSET' value: <i>float</i> entre -10 e 10 V	⑧
Driver		
Set	'DAC(i):WG:OFFSET:(Instrument Value)##' i: índice do DAC Instrument Value: <i>float</i> entre -10 e 10 V ex: 'DAC1:WG:OFFSET:-5.0##'	⑥
Get	'DAC(i):WG:OFFSET?##' i: índice do DAC ex: 'DAC1:WG:OFFSET?##'	③
Plataforma		
Set	Set_DAC_WG_Offset(DAC *d, int id, float Value) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC float Value: <i>float</i> entre -10 e 10 V	⑦
Get	float Value =Get_DAC_WG_Offset(DAC *d, int id) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC	④
	'(Instrument_Value)##' Instrument_Value: <i>float</i> entre -10 e 10 V	⑤

Para configurar o offset do sinal gerado diretamente através dos métodos, o utilizador da *toolbox* deve especificar um valor do tipo *float* entre -10 e 10 V.

### Duty Cycle da onda gerada

O *duty cycle* do sinal gerado pode ser definido na interface gráfica desta *toolbox* através do painel apresentado na Figura 4-20 ou utilizando os métodos descritos na Tabela 4-21.



Figura 4-20 - Painel Duty Cycle do separador Wave Generator.

Tabela 4-21 - Métodos WG\_Duty.

GUI		
Callback	WG1_Duty_ValueChanged () i: índice do WG	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_DUTY' PropertyValue: <i>float</i> de 0 a 100%	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'WG_DUTY' value: <i>float</i> de 0 a 100%	⑧
Driver		
Set	'DAC(i):WG:DUTY:(Instrument Value)##' i: índice do DAC Instrument Value: <i>float</i> de 0 a 100% ex: 'DAC1:WG:DUTY:75.0##'	⑥
Get	'DAC(i):WG:DUTY?##' i: índice do DAC ex: 'DAC1:WG:DUTY?##'	③
Plataforma		
Set	Set_DAC_WG_Duty(DAC *d, int id, float Value) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC float Value: <i>float</i> de 0 a 100%	⑦
Get	float Value =Get_DAC_WG_Duty(DAC *d, int id) DAC *d: Apontador para a estrutura do DAC int id: Índice do DAC	④
	'(Instrument_Value)##' Instrument_Value: <i>float</i> de 0 a 100%	⑤

O *duty cycle* da onda selecionado deve ser um valor do tipo *float* entre 0 e 100%.



### 4.3.3. Interruptores Analógicos

A necessidade de um corte rápido na atuação ou alteração dos sinais aplicados ao DUT (*Device Under Test*) em alguns testes de MEMS levou à integração de um bloco de interruptores analógicos de controlo digital TTL na GUI. O seu painel de configuração é apresentado na Figura 4-21.

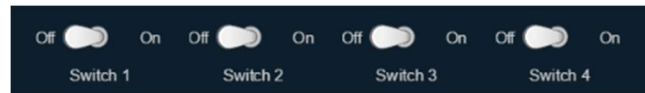


Figura 4-21 - Painel dos interruptores analógicos.

Para possibilitar o controlo destes interruptores, foram desenvolvidos os métodos listados na Tabela 4-22, baseados no modelo de configuração descrito na secção 4.1.

Tabela 4-22 - Métodos Switch.

GUI		
Callback	Switch(i)_ValueChanged () i: índice do interruptor	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'SWITCH' PropertyValue: 1 ou 0	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.DAC1) 'PropertyName': 'SWITCH' value: 1 ou 0	⑧
Driver		
Set	'SW:(i):(Instrument Value)##' i: índice do interruptor: 1 a 4 Instrument Value: 1 ou 0 ex: 'SW:2:1##'	⑥
Get	'SW:(i)?##' i: índice do interruptor: 1 a 4 ex: 'SW:2?##'	③
Plataforma		
Set	Set_SW_State(int sw, int Value) int sw: Índice do interruptor: 1 a 4 int Value: 1 ou 0	⑦
Get	int Value =Get_SW_STATE(int sw) int sw: Índice do interruptor: 1 a 4	④
	'(Instrument_Value)##' Instrument_Value: 1 ou 0	⑤

Para ativar ou desativar os interruptores analógicos, o valor enviado deverá ser um valor inteiro, 1 ou 0, respetivamente.

## 4.4. Painel Lock-In

Uma das principais características desta plataforma de teste de MEMS é a inclusão de um amplificador de *lock-in* digital de ganho variável, permitindo a medição de pequenos sinais AC de apenas alguns nanoVolts, mesmo na presença de fontes de ruído de muito maior amplitude. A arquitetura deste mecanismo pode ser visualizada na Figura 4-22.

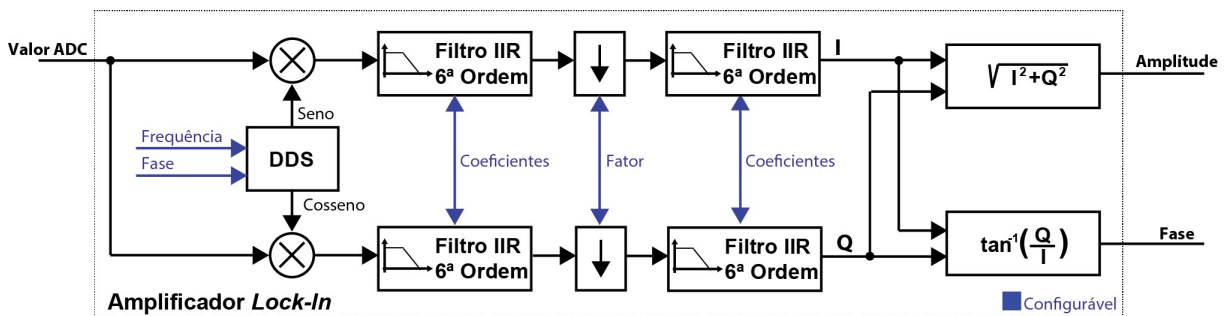


Figura 4-22 - Arquitetura do amplificador de *lock-in*.

A configuração utilizada neste amplificador possibilita ao utilizador configurar os filtros, o fator de decimação, a frequência e o ganho do sinal. De modo a permitir ao utilizador configurar estas propriedades a partir da interface gráfica, foi desenvolvido o painel “Lock-In Configuration” apresentado na Figura 4-23. As opções de configuração e os métodos que garantem o seu funcionamento são descritos nas subsecções seguintes.

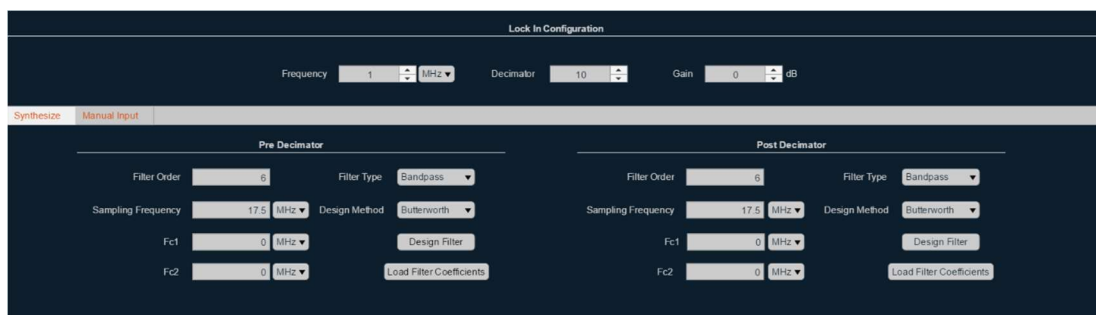


Figura 4-23 - Painel Lock-in Configuration.

### 4.4.1. Configuração da frequência

De forma a definir a frequência do sinal de entrada do DDS (*Direct Digital Synthesizer*), foi criado, na interface gráfica, o painel “Frequency”, podendo este ser visualizado na Figura 4-24.



Figura 4-24 - Painel “Frequency” do separador Lock-in.

Este painel utiliza os métodos elencados na Tabela 4-23 para efetuar a configuração do valor de frequência pretendido, aceitando valores de frequência do tipo *float* até 17,5 MHz.

Tabela 4-23 - Métodos LI\_Freq.

GUI		
Callback	LI1_Freq_ValueChanged () i: índice do Lock In	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.LI1) 'PropertyName': 'FREQ' PropertyValue: <i>float</i> positivo até 17,5 MHz	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.LI1) 'PropertyName': 'FREQ' value: <i>float</i> positivo até 17,5 MHz	⑧
Driver		
Set	'LI(i):FREQ:(Instrument Value)##' i: índice do Lock In Instrument Value: <i>float</i> positivo até 17,5 MHz ex: 'LI1:FREQ:150000##'	⑥
Get	'LI(i):FREQ?##' i: índice do Lock In ex: 'LI1:FREQ?##'	③
Plataforma		
Set	Set_LI_Freq(LOCK_IN *d, <i>float</i> Value) LOCK_IN *d: Apontador para a estrutura do Lock In <i>float</i> Value: <i>float</i> positivo até 17,5 MHz	⑦
Get	<i>float</i> Value =Get_LI_Freq(LOCK_IN *d) LOCK_IN *d: Apontador para a estrutura do Lock In '(Instrument_Value)##' Instrument_Value: <i>float</i> positivo até 17,5 MHz	④ ⑤

## 4.4.2. Configuração do ganho

Para além da configuração da frequência do DDS, esta *toolbox* permite também definir o ganho para a geração de portadoras do *lock-in*, valendo-se do painel “Gain” apresentado na Figura 4-25 e utilizando os métodos descritos na Tabela 4-24.



Figura 4-25 - Painel Gain do separador Lock-in.

Os valores permitidos para a definição deste ganho são *floats* entre -6 e 26 dB.

Tabela 4-24 - Métodos LI\_Gain.

GUI		
Callback	LI1_Gain_ValueChanged () i: índice do Lock In	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.LI1) 'PropertyName': 'GAIN' PropertyValue: <i>float</i> entre -6 e 26 dB	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.LI1) 'PropertyName': 'GAIN' value: <i>float</i> entre -6 e 26 dB	⑧
Driver		
Set	'LI(i):GAIN:(Instrument Value)##' i: índice do Lock In Instrument Value: <i>float</i> entre -6 e 26 dB ex: 'LI1:GAIN:10##'	⑥
Get	'LI(i):GAIN?##' i: índice do Lock In ex: 'LI1:GAIN?##'	③
Plataforma		
Set	Set_LI_Gain(LOCK_IN *d, <i>float</i> Value) LOCK_IN *d: Apontador para a estrutura do Lock In <i>float</i> Value: <i>float</i> entre -6 e 26 dB	⑦
Get	<i>float</i> Value =Get_LI_Gain(LOCK_IN *d) LOCK_IN *d: Apontador para a estrutura do Lock In	④
	'(Instrument_Value)##' Instrument_Value: <i>float</i> entre -6 e 26 dB	⑤

### 4.4.3. Configuração do filtro decimador

Tal como pode ser visualizado na Figura 4-22, a arquitetura do amplificador de *lock-in* possui um filtro decimador entre os dois níveis de filtragem. Sendo que o segundo filtro é usado para definir a largura de banda do sinal modulado, e considerando a impossibilidade de este ser estável para baixas frequências dada a elevada frequência de amostragem do ADC, o filtro decimador permite tornar este filtro estável efetuando um *downsampling* do sinal. O painel “Decimator” elencado na Figura 4-26 permite a configuração deste valor de decimação.



Figura 4-26 - Painel Decimator do separador Lock-in.

O mecanismo de configuração deste parâmetro é definido pelos fluxogramas apresentados na Figura 4-2 e na Figura 4-3, utiliza os métodos definidos na Tabela 4-25 e pode ser definido utilizando valores *float* positivos até 1000000.

Tabela 4-25 - Métodos LI\_Dec.

GUI		
Callback	LI1_Dec_ValueChanged () i: índice do Lock In	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.LI1) 'PropertyName': 'DECIMATOR' PropertyValue: <i>float</i> positivo até 1000000	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.LI1) 'PropertyName': 'DECIMATOR' value: <i>float</i> positivo até 100000	⑧
Driver		
Set	'LI(i):DEC:(Instrument Value)##' i: índice do Lock In Instrument Value: <i>float</i> positive até 1000000 ex: 'LI1:DEC:10##'	⑥
Get	'LI(i):DEC?##' i: índice do Lock In ex: 'LI1:DEC?##'	③
Plataforma		
Set	Set_LI_Dec(LOCK_IN *d, <i>float</i> Value) LOCK_IN *d: Apontador para a estrutura do Lock In <i>float</i> Value: <i>float</i> positivo até 1000000	⑦
Get	<i>float</i> Value =Get_LI_Dec(LOCK_IN *d) LOCK_IN *d: Apontador para a estrutura do Lock In	④
	'(Instrument_Value)##' Instrument_Value: <i>float</i> positivo até 1000000	⑤

#### 4.4.4. Sintetização de filtros

A arquitetura deste amplificador de *lock-in* inclui dois níveis de filtragem de forma a permitir a extração da amplitude e fase do sinal. Estes filtros IIR de 6ª ordem estão divididos em três secções de segunda ordem e têm como objetivo a remoção da portadora e do ruído do sinal adquirido.

Pretendendo dotar o utilizador da *toolbox* com a capacidade de dimensionar estes filtros, foi criado o painel “Synthesize” na GUI, elencado na Figura 4-27.

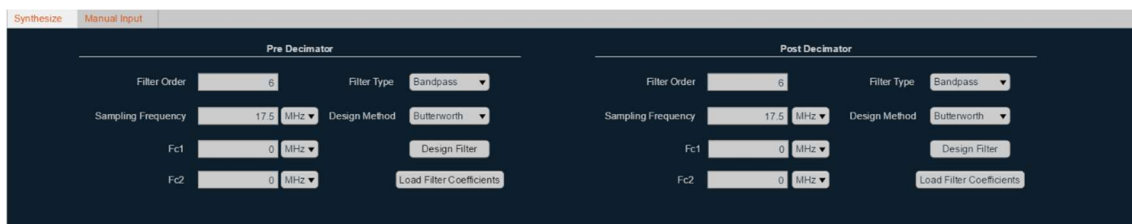


Figura 4-27 - Painel Synthesize do separador Lock-in.

Para permitir a sintetização do filtro desejado na própria aplicação, implementou-se neste painel um conjunto de configurações que, com recurso ao Filter Designer [23] do MATLAB, permitem dimensionar o filtro implementado em *hardware*. Esta interface permite o desenho de filtros *Butterworth*, podendo estes ser do tipo:

- Passa-baixo;
- Passa-alto;
- Passa-banda;
- Rejeita-banda.

Para isto, o utilizador deve definir a frequência de amostragem desejada, assim como as frequências de corte do filtro desejado,  $F_{3dB}$  para filtros do tipo passa-baixo e passa-alto e  $F_{c1}$  e  $F_{c2}$  para passa-banda e rejeita-banda. Depois de seleccionadas as configurações desejadas, pressionando o botão “Design Filter”, é aberta uma janela do Filter Visualization Tool, demonstrada na Figura 4-28, onde é possível visualizar a resposta do filtro em fase e amplitude, entre outras informações relevantes, incluindo os seus coeficientes.

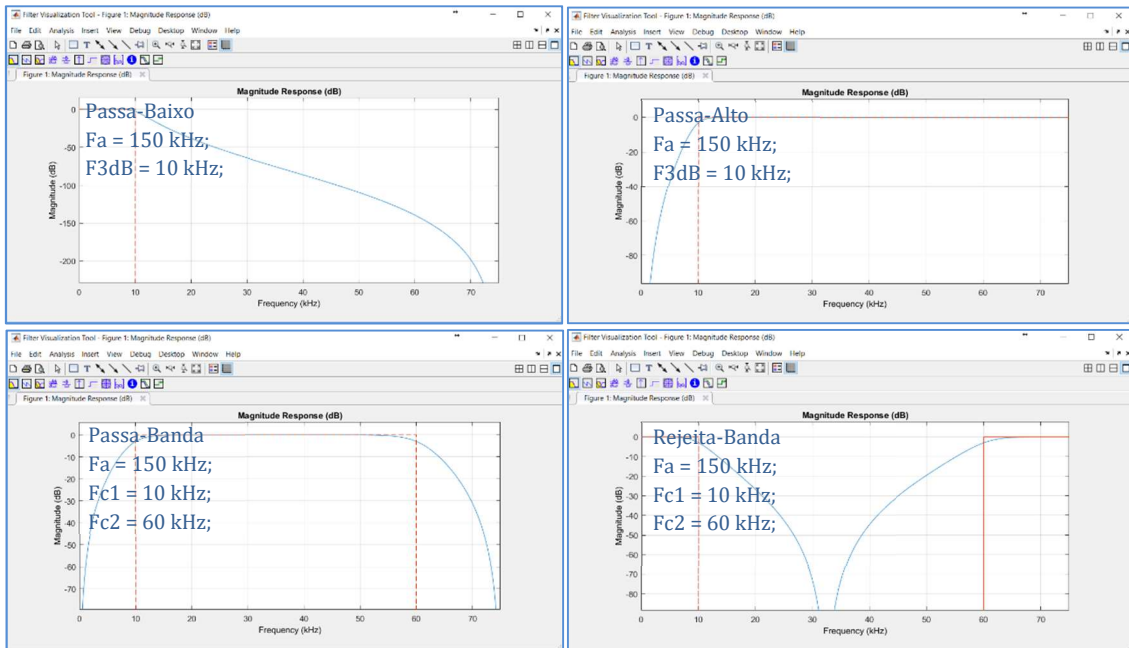


Figura 4-28 - Exemplo de janelas Filter Visualization Tool para os quatro tipos de filtro.

Caso o filtro dimensionado esteja como desejado, o utilizador pode premir o botão “Load Filter Coefficients”, sendo os coeficientes do filtro dimensionado automaticamente carregados para a plataforma, valendo-se para isso dos mecanismos *set* e *get*, desenvolvidos nesta *toolbox*. Os métodos utilizados para configurar os coeficientes do filtro pré decimação e pós decimação podem ser analisados na Tabela 4-26.



Figura 4-29 - Painel Manual Input do separador Lock-in.

A interface gráfica possibilita também a configuração manual dos coeficientes através do separador “Manual Input” demonstrado na Figura 4-29. Os métodos utilizados para definir os coeficientes dos filtros, por via manual ou automática, são descritos na Tabela 4-26.

Tabela 4-26 - Métodos de configuração dos coeficientes.

GUI	
Callback	(f)_s)_c)_ValueChanged () f: filtro (Pre, Pos) s: secção do filtro (1,2,3) c: nome do coeficiente (A0,A1,A2,B1,B2,G) ①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.LI1) 'PropertyName': 'POST_S(s)_c)' ou 'PRE_S(s)_c)' s: secção do filtro (1,2,3) c: nome do coeficiente (A0,A1,A2,B1,B2,G) PropertyValue: <i>float</i> de $-\infty$ a $\infty$ ②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.LI1) 'PropertyName': 'POST_S(s)_c)' ou 'PRE_S(s)_c)' value: <i>float</i> de $-\infty$ a $\infty$ ⑧
Driver	
Set	'LI(i):(F):(c):(s):(Instrument Value)##' i: índice do Lock In F: filtro (PRE, POS) s: secção do filtro (1,2,3) c: nome do coeficiente (A0,A1,A2,B1,B2,G) Instrument Value: <i>float</i> de $-\infty$ a $\infty$ ex: 'LI1:POST:A1:2:0.5234##' ⑥
Get	'LI(i):(F):(c):(s)?' i: índice do Lock In F: filtro (PRE, POS) s: secção do filtro (1,2,3) c: nome do coeficiente (A0,A1,A2,B1,B2,G) ex: 'LI1:PRE:A1:2?' ③
Plataforma	
Set	Set_LI_(F)_c)(LOCK_IN *d, int id, float Value) F: filtro (PRE, POS) c: nome do coeficiente (A0,A1,A2,B1,B2,G) LOCK_IN *d: Apontador para a estrutura do Lock In int id: secção do filtro float Value: <i>float</i> de $-\infty$ a $\infty$ ⑦
Get	<i>float</i> Value =Get_LI_(F)_c)(LOCK_IN *d, int id) F: filtro (PRE, POS) c: nome do coeficiente (A0,A1,A2,B1,B2,G) LOCK_IN *d: Apontador para a estrutura do <i>Lock In</i> int id: secção do filtro ④
	'(Instrument_Value)##' Instrument_Value: <i>float</i> de $-\infty$ a $\infty$ ⑤



## 4.5. Painel Acquire

Com o objetivo de possibilitar a visualização da resposta da estrutura sob teste, foi desenvolvido o painel “Acquire”, apresentado na Figura 4-30. Este painel permite ao utilizador requerer à plataforma o número desejado de amostras, especificando a frequência de amostragem e o tipo de sinal de saída ou determinando diretamente o número de amostras. Estes valores são então enviados para a interface gráfica e através da porta 8 do TCP/IP, sendo apresentados neste mesmo painel, quando todas as amostras forem recebidas. Este modo de aquisição permite fazer leituras com frequências de amostragem altas, na ordem dos MHz, sem a perda de amostras e sem comprometer o normal funcionamento de todo o sistema.

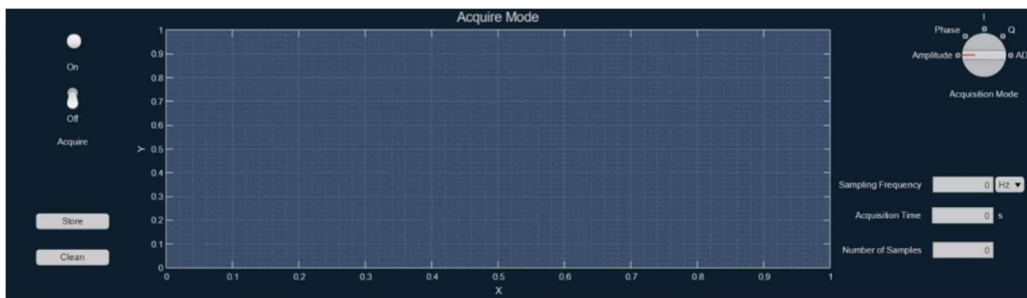


Figura 4-30 - Painel Acquire.

Utilizando o *dial* “Display Mode”, demonstrado na Figura 4-31, é possível selecionar entre os vários tipos de dados a adquirir, mais especificamente:

- Amplitude;
- Fase;
- I (In-phase Component);
- Q (Quadrature Component);
- Multiplicação;
- ADC;
- Saída filtrada.

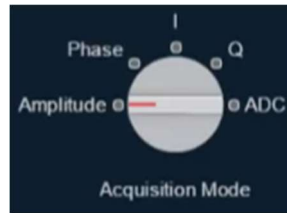


Figura 4-31 - Dial de seleção do modo de aquisição.

O utilizador da interface gráfica tem também a opção de guardar os dados adquiridos para análise, recorrendo ao botão “Store”, que permite o arquivo dos dados adquiridos diretamente no *workspace* do MATLAB.

O pedido de dados através deste modo de aquisição é efetuado com recurso aos valores da Tabela 4-28, valendo-se dos métodos expostos na Tabela 4-27.

Tabela 4-27- Métodos do modo Acquire.

GUI		
Callback	AcquireButton_ValueChanged ()	①
Invoke	invoke(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.ACQ1) 'PropertyName': Tabela 4-28 PropertyValue: Número de amostras	⑨
Driver		
Invoke	'GET:(M):(Instrument Value)##' M: Tabela 4-28 Instrument Value: Número de amostras ex: 'DAC1:SEL:WG##'	⑩

Tabela 4-28 - Valores para seleção do modo de aquisição.

Property Value	Instrument Value
Amplitude	AMP
Phase	PHS
I	I
Q	Q
ADC	ADC
Filter	FILTER
Multiplication	MULT

### 4.5.1. Seleção da frequência de amostragem

A configuração da frequência de amostragem dos dados a receber é definida através do painel “Sampling Frequency”, podendo este ser visualizado na Figura 4-32. Esta configuração recorre aos métodos *set* e *get* descritos na secção inicial deste capítulo, estando os seus métodos expostos na Tabela 4-29.



Figura 4-32 - Painel de configuração da frequência de amostragem.

A frequência de amostragem pode ser definida com recurso a *floats* até 3,5 MHz.

Tabela 4-29 - Métodos Sampling Frequency.

GUI		
Callback	SamplingFrequency_ValueChanged ()	①
Set	set(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.ACQ1) 'PropertyName': 'SAMPLING_FREQ' PropertyValue: <i>float</i> até 3.5 MHz	②
Get	value=get(H,'PropertyName'); H: Handle (ex. App.deviceObj.ACQ1) 'PropertyName': 'Sampling_FREQ' value: <i>float</i> até 3.5 MHz	⑧
Driver		
Set	'SET:SFREQ:(Instrument Value)##' Instrument Value: <i>float</i> até 3.5 MHz ex: 'SET:SFREQ:10000##'	⑥
Get	'SET:SFREQ?##' ex: 'SET:SFREQ?##'	③
Plataforma		
Set	Set_SFreq( <i>float</i> Value) <i>float</i> Value: <i>float</i> até 3.5 MHz	⑦
Get	<i>float</i> Value =Get_SFreq() <i>float</i> Value: <i>float</i> até 3.5 MHz	④
	'(Instrument_Value)##' Instrument_Value: <i>float</i> até 3.5 MHz	⑤

## 4.6. Painel Live Mode

Analogamente ao modo de aquisição apresentado na secção anterior, o painel Live Mode foi desenvolvido para permitir a receção de dados vindos da plataforma, com o apanágio de disponibilizar a sua visualização em direto. O utilizador seleciona novamente a frequência de amostragem e tipo de sinal a receber, iniciando-se a receção de dados quando o *switch* “Live Mode”, representado na Figura 4-34, for alterado para “On”. Este envio mantém-se até que o utilizador retorne o *switch* à posição “Off”. Este modo tem a vantagem de permitir ao utilizador da plataforma configurar os parâmetros desejados e ver em tempo real as alterações na resposta da estrutura.

Este painel, que pode ser visualizado na Figura 4-33, disponibiliza funcionalidades conformes com as apresentadas na secção anterior, ao nível da seleção do modo de aquisição e permitindo para este caso o arquivo no *workspace* do MATLAB dos últimos cinco milhões de amostras recebidas. Os seus métodos de funcionamento podem ser visualizados na Tabela 4-30 e os seus parâmetros de entrada na Tabela 4-31.



Figura 4-33 - Painel Live Mode.

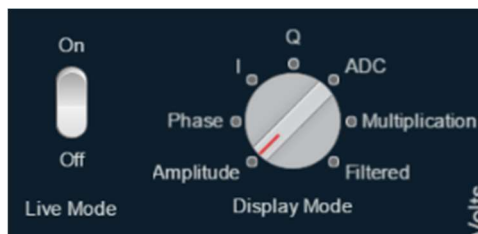


Figura 4-34 - Botão Live Mode e dial Display Mode.

Através do *dial* “Display Mode”, demonstrado na Figura 4-34, é possível selecionar entre os vários tipos de dados a adquirir, sendo estes:

- Amplitude;
- Fase;
- I (In-phase Component);
- Q (Quadrature Component);
- Multiplicação;
- ADC;
- Saída filtrada.

Tabela 4-30-Métodos Live Mode.

GUI	
Callback	Live_Mode_Enable_ValueChanged () <span style="float: right;">①</span>
Invoke	invoke(H,'PropertyName',PropertyValue); H: Handle (ex. App.deviceObj.ACQ1) <span style="float: right;">⑨</span> 'PropertyName': Tabela 4-31 PropertyValue: Frequência de amostragem ( <i>float</i> até 17.5 MHz)
Driver	
Invoke	'LIVE:(Instrument Value)##' Instrument Value: Tabela 4-31 <span style="float: right;">⑩</span> ex: 'LIVE:AMP##'  'LIVE:STOP##' Mensagem de fim de aquisição

Tabela 4-31 - Valores para seleção do modo de aquisição do Live Mode.

Property Value	Instrument Value
Amplitude	AMP
Phase	PHS
I	I
Q	Q
ADC	ADC
Filter	FILTER
Multiplication	MULT

## 4.7. Controlos de visualização

Para além destas funcionalidades e com a inerente necessidade de controlar as propriedades de visualização destes dados em tempo real, este painel beneficia da capacidade de seleção entre controlo automático e manual do eixo Y e controlo manual do eixo temporal. Estes mecanismos manuais desenvolvidos no âmbito desta *toolbox* podem ser visualizados na Figura 4-35.

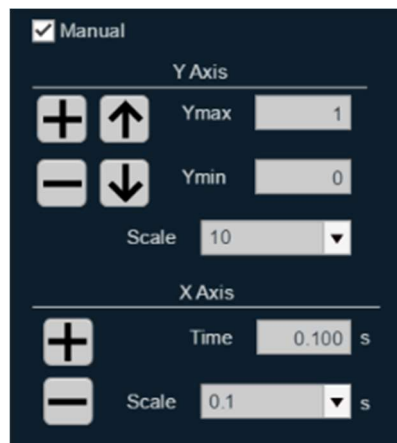


Figura 4-35 - Painel de controlos manuais de visualização.

Relativamente ao eixo Y, quando é efetuada a seleção do controlo manual, as características do modo automático são mantidas, sendo então possível definir os limites máximo e mínimo deste eixo por introdução dos valores desejados, *zoom in* e *zoom out*, assim como a deslocação de ambos os eixos uniformemente segundo o valor definido na caixa “*Scale*”.

Para o eixo X, é possível definir o tempo amostrado por divisão na caixa de “*Time Scale*”, assim como aumentar ou diminuir este valor através dos botões “*zoom in*” e “*zoom out*” em conjunto com a componente “*Scale*”.

## 4.8. Painel de configuração da comunicação

O painel elencado na Figura 4-36 permite estabelecer a comunicação com o *driver* de comunicação e conseqüentemente com a plataforma de teste de MEMS, utilizando o IP introduzido na caixa de texto, através das portas 7 e 8 do TCP/IP.

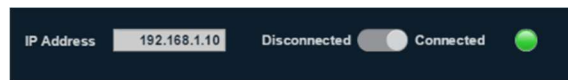


Figura 4-36 - Painel de configuração da comunicação.

## 4.9. Painel Pull-in

Aplicando uma tensão nos elétrodos paralelos de um atuador electrostático, a força electrostática resultante irá provocar um deslocamento, de forma a diminuir a distância entre as placas do condensador. Se a tensão aplicada for baixa, a força elástica da mola irá contrariar esta força eletrostática, encontrando um ponto de equilíbrio. Esta característica altera-se, no entanto, à medida que a tensão de atuação aumenta, deixando a força elástica de ser suficiente para manter o equilíbrio com a força eletrostática, fazendo com que a distância entre as placas do condensador diminua rapidamente até estas se encostarem. Este fenómeno é denominado *pull-in*, sendo a tensão necessária para que este ocorra a tensão de *pull-in* [24]. Inversamente, a tensão máxima de atuação, a partir da qual as placas do condensador, partindo da posição de *pull-in*, se desencostam restaurando o equilíbrio entre a força electrostática e a força elástica, é denominada tensão de *pull-out*. Estas características são particularmente interessantes quando aliadas à medição da frequência de ressonância da estrutura, permitindo estimar as não idealidades do processo de fabrico, como é o caso do *over-etching* que consiste na remoção em excesso de material no fabrico da estrutura [25].

Com o objetivo de medir estas propriedades foi desenvolvido o painel apresentado na Figura 4-37, juntamente com os métodos definidos na Tabela 4-32.

#### 4. Funcionalidades da *Toolbox*

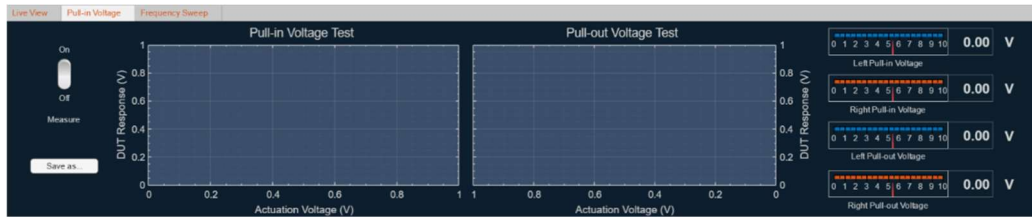


Figura 4-37 - Painel Pull-in.

De forma a obter a tensão de *pull-in*, são efetuados dois testes, um de cada lado da estrutura diferencial, com recurso aos DACs incluídos na plataforma de teste. Para cada lado, é aplicada uma rampa, com início em 1 V e incrementos de 10 mV, até que seja detetada uma variação superior ao *threshold* definido ou até 10 V, no caso de não ser detetado *pull-in*. No momento em que o *pull-in* é detetado, a atuação é desativada com recurso aos interruptores da plataforma, de modo a garantir que a estrutura retorna à sua posição de equilíbrio. O algoritmo de medição da tensão de *pull-out* tem um princípio de funcionamento semelhante, sendo novamente efetuados dois testes, um para cada lado da estrutura. O teste inicia com uma tensão de atuação igual à tensão de *pull-in* detetada previamente para esse lado da estrutura. Esse valor é então decrementado com intervalos de 10 mV até que seja detetada uma variação superior ao *threshold* mencionado na deteção do *pull-in*, desta vez no sentido inverso. Esta tensão de atuação é decrementada até 0 V caso não seja detetado *pull-out*.

Esta rotina de teste utiliza o método *invoke* descrito na secção 4.1 e os seus métodos são descritos na Tabela 4-32.

Tabela 4-32 - Métodos da rotina de medição da tensão de *pull-in*.

GUI		
Callback	PIV_Switch_ValueChanged ()	①
Invoke	invoke(H,'PropertyName'); H: Handle (ex. App.deviceObj.ACQ1) 'PropertyName': 'Pullin_Voltage'	⑨
Driver		
Invoke	'GET:PIV##'	⑩



## 4.10. Painel Frequency Sweep

Uma rotina particularmente interessante no teste de estruturas MEMS é o varrimento de frequências através do qual é possível extrair várias características do sistema em teste, tal como a frequência de ressonância e o fator de qualidade. Assim, foi incluído nesta *toolbox* um mecanismo desta natureza, permitindo a geração de duas ondas sinusoidais desfasadas 180° entre si, de forma a atuarem os dois lados da estrutura em sentidos inversos, sendo configuráveis as seguintes características:

- Frequência inicial;
- Frequência final;
- Incremento de frequência;
- Número de amostras por frequência;
- Amplitude do sinal aplicado;
- *Offset* das sinusoides.

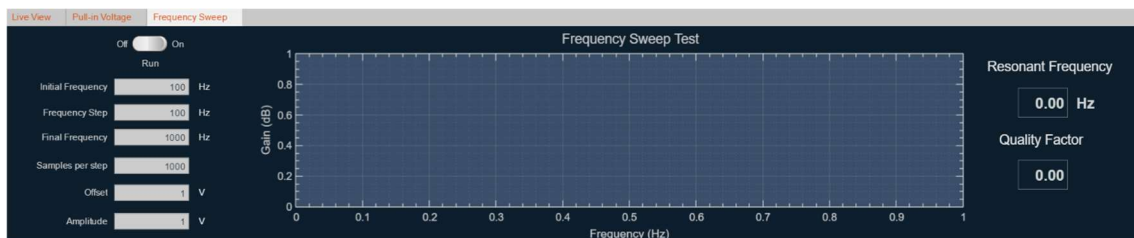


Figura 4-38 - Painel Frequency Sweep.

O painel “Frequency Sweep”, ilustrado na Figura 4-38, permite a configuração destas características, assim como a visualização dos dados adquiridos na execução desta rotina, relativos à resposta da estrutura sob teste. Este painel permite ainda a medição da frequência de ressonância e do fator de qualidade do DUT, utilizando um detetor de pico em conjunto com as funções de *fit postprocessing* da *Curve Fitting Toolbox* [25] do *MATLAB*.

Os métodos utilizados nesta rotina de varrimento de frequências são descritos na Tabela 4-33.

Tabela 4-33 - Métodos da rotina de varrimento de frequências.

GUI	
Callback	FS_Switch_ValueChanged () <span style="float: right;">①</span>
Invoke	invoke(H,'PropertyName',FS_initial, FS_step, FS_final, Samples, offset, amplitude); H: Handle (ex. App.deviceObj.ACQ1) <span style="float: right;">⑨</span> 'PropertyName': 'Frequency_Sweep' FS_initial: Frequência inicial ( <i>float</i> positivo até 17.5 MHz) FS_step: Incremento de frequência ( <i>float</i> positivo) FS_final: Frequência final ( <i>float</i> positivo até 17.5 MHz) Samples: Numero de amostras por frequência ( <i>float</i> positivo) offset: Offset das sinusoides ( <i>float</i> -10V a 10V) amplitude: Amplitude das sinusoides ( <i>float</i> -10V a 10V)
Driver	
Invoke	'FSWEEP:(FS_i):(FS_s):(FS_f):(S):( o):(a)##' <span style="float: right;">⑩</span> FS_i: Frequência inicial ( <i>float</i> positivo até 17.5 MHz) FS_s: Incremento de frequência ( <i>float</i> positivo) FS_f: Frequência final ( <i>float</i> positivo até 17.5 MHz) S: Número de amostras por frequência ( <i>float</i> positivo) o: Offset das sinusoides ( <i>float</i> -10V a 10V) a: Amplitude das sinusoides ( <i>float</i> -10V a 10V)

## 4.11. Conclusão do capítulo

Neste capítulo, foram abordadas as funcionalidades da *toolbox* desenvolvida no âmbito desta dissertação, descrevendo o funcionamento e modos de utilização de cada um dos mecanismos de configuração, aquisição e teste implementados.

No próximo capítulo, serão apresentados os resultados de algumas das funcionalidades descritas neste capítulo.

# 5.

# Resultados

Neste capítulo, são apresentados os resultados do desenvolvimento desta dissertação. Em primeiro lugar, será analisado o estado do funcionamento das configurações, recorrendo para isso a algumas figuras que ilustram o resultado da configuração efetuada. Em seguida, será demonstrado o resultado da execução dos processos de aquisição de dados, referindo as especificações atingidas com o sistema desenvolvido e apresentando testemunhos visuais do mesmo. Para finalizar, serão analisados os resultados obtidos nos testes específicos de estruturas MEMS, com análise de gráficos e imagens obtidos da interface gráfica desenvolvida nesta dissertação.

## 5.1. Resultados das configurações

Como resultado da criação dos painéis de configuração, foi possível garantir a correta alteração de todos os parâmetros relativos aos canais de leitura e atuação e amplificador de *lock-in*. Nas figuras seguintes, podem ser visualizados alguns exemplos dessas alterações, recorrendo para isso ao painel “Live Mode” desta toolbox, definido para apresentar a resposta em amplitude do DUT. Para este exemplo foram configuradas, com recurso à interface gráfica, duas ondas quadradas de 100 Hz geradas pelos DACs incluídos na plataforma, com 1 V de *offset* e amplitude e desfasadas 180°, sendo cada uma delas aplicada a cada um dos lados da estrutura.

A Figura 5-1 demonstra a alteração do fator de decimação do amplificador de *lock-in*, sendo este definido em a) como um valor de 10 e em b) como um valor de 100, através da utilização do painel “Lock-In Configuration”.



Figura 5-1 - Resultados da configuração do filtro decimador do amplificador de *lock-in*.



Figura 5-2 - Resultados da configuração do ganho do amplificador de *lock-in*.

Na Figura 5-2, pode ser observado o resultado da alteração do valor do ganho do amplificador de *lock-in*, tendo sido utilizado para este efeito o painel “Lock-In Configuration”, sendo a) a definição desta propriedade como 1 dB e b) a sua configuração como 3 dB.

A Figura 5-3 demonstra a resposta da estrutura sob teste a duas ondas triangulares geradas pelos DACs. Estes sinais de 100 Hz e 0,1 V de amplitude e *offset* são aplicados cada um a um lado do acelerómetro utilizado e têm fases de  $0^\circ$  e  $180^\circ$ , respetivamente.

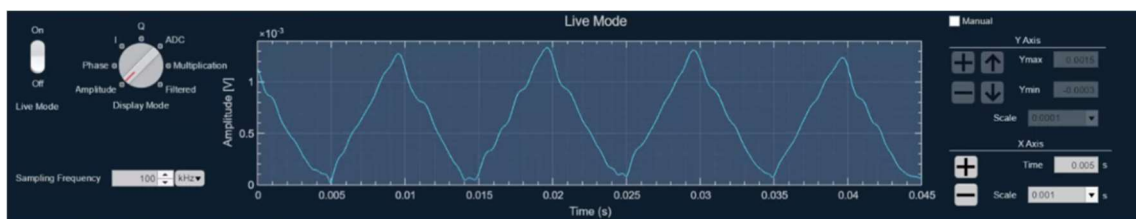


Figura 5-3 - Resultado da configuração do gerador de ondas com uma onda triangular.

## 5.2. Resultados dos blocos de aquisição

Com o desenvolvimento dos blocos de aquisição, foi possível desenvolver mecanismos que garantam a recepção e visualização dos sinais provenientes do sistema de leitura da plataforma de teste de MEMS. Nas seguintes subsecções, são apresentados os resultados da criação destes blocos, utilizando, para este efeito, capturas dos respetivos painéis da interface gráfica.

### 5.2.1. Acquire Mode

O bloco “Acquire Mode” desenvolvido permite a aquisição de sinais provenientes do sistema de leitura com frequência de amostragem até 3,5 MHz, limitado por *software*. No entanto, durante a aquisição de dados através deste mecanismo, não é possível comunicar com a plataforma. A visualização destes dados é disponibilizada com recurso ao seu painel da GUI, após a conclusão da aquisição, sendo também possível a sua análise no MATLAB, efetuando a sua gravação no *workspace* através do botão “Store”.

A Figura 5-4 apresenta alguns resultados do desenvolvimento deste mecanismo, demonstrando a aquisição da resposta em amplitude da estrutura sob teste a uma onda sinusoidal, com uma frequência de amostragem de 100 kHz durante:

- a) 50 ms (5000 amostras);
- b) 500 ms (50.000 amostras);
- c) 1 s (100.000 amostras);
- d) 10 s (1.000.000 de amostras).

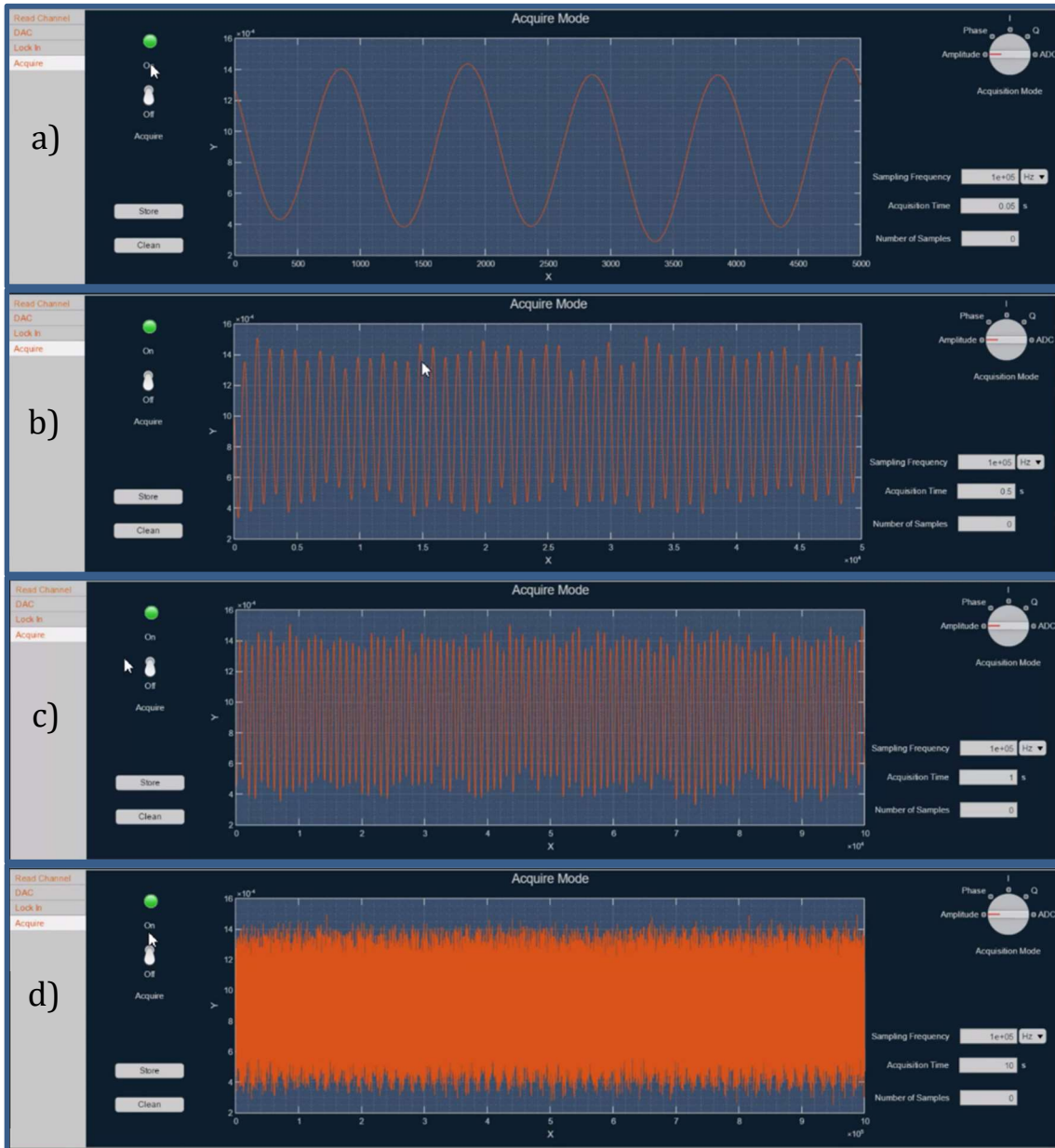


Figura 5-4 - Resultados do modo de aquisição para diferentes durações.

## 5.2.2. Live Mode

O segundo método de aquisição desenvolvido no âmbito desta *toolbox* é o “Live Mode”. Como resultado da implementação deste, é possível obter dados da plataforma de teste MEMS em tempo real, permitindo assim a configuração simultânea das características deste sistema de testes e a visualização dos efeitos destas alterações na resposta do DUT. O mecanismo “Live Mode” possibilita a aquisição dos tipos de resposta enumerados na secção 4.6 deste documento, com frequências de amostragem até 1 MHz e uma largura de janela máxima de 2,5 s, sem comprometer o normal funcionamento da interface gráfica e garantindo a fluidez dos dados apresentados. Permite também ao utilizador efetuar um pedido através do “Acquire Mode”, enquanto a receção de dados em tempo real ainda se encontra ativa. Neste caso, a receção de dados em direto é pausada durante a aquisição dos sinais pelo “Acquire Mode”, sendo retomada em seguida.

Na Figura 5-5, é possível visualizar uma captura de ecrã da interface gráfica que demonstra ambos os painéis, sendo demonstrado no painel superior a resposta do DUT a uma onda sinusoidal em tempo real e no painel inferior o mesmo sinal adquirido com o “Acquire Mode”.



Figura 5-5 - Resultados do funcionamento do "Live Mode" e "Acquire Mode".



### 5.3. Resultados das rotinas de teste de MEMS

As rotinas de teste de MEMS desenvolvidas nesta dissertação permitem a caracterização de algumas propriedades de elevada relevância para o teste deste tipo de estruturas. Na Figura 5-6, é apresentado um exemplo de testes incluído no documento da Embedded MEMS Platform for structure test and characterization [7] descrita na secção 2.5, e de funcionamento idêntico ao sistema desenvolvido nesta dissertação. Estes testes, efetuados a algumas estruturas semelhantes às utilizadas para elaboração dos testes da *toolbox* desenvolvida, apresentam valores de tensão de *pull-in*, frequência de ressonância e fator de qualidade (Q).

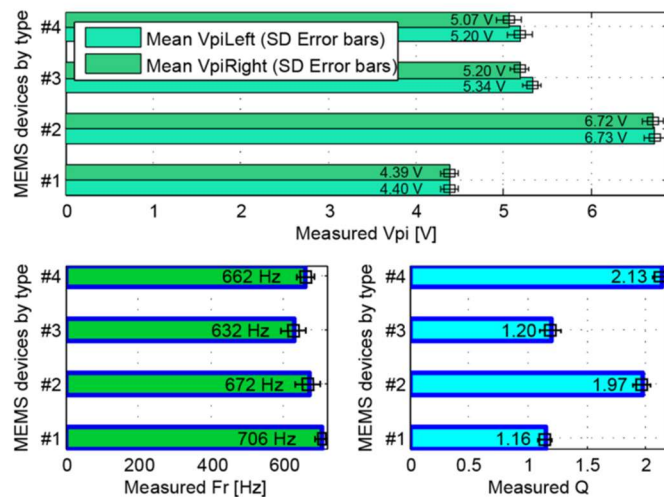


Figura 5-6 - Resultados da caracterização de acelerómetros MEMS capacitivos da Embedded MEMS Platform for structure test and characterization [7].

Nas subsecções seguintes, serão apresentados os resultados do desenvolvimento destes blocos, recorrendo para esse efeito a capturas de ecrã dos seus painéis na interface gráfica e a gráficos criados para análise dos resultados adquiridos.

### 5.3.1. Tensões de *pull-in* e *pull-out*

As rotinas de medição de tensões de *pull-in* e *pull-out*, abordadas na secção 4.9 deste documento, permitem a medição destas propriedades em dispositivos MEMS *parallel-plate* em que estas características se encontrem abaixo dos 10 V, devido às limitações do algoritmo e do sistema de atuação das estruturas sob teste. De forma a validar o funcionamento deste mecanismo, foram efetuados testes recorrendo a um acelerómetro compatível com as restrições mencionadas acima.

Nas seguintes figuras, são apresentados os resultados destas medições, podendo ser observadas na Figura 5-7 as capturas de ecrã da interface gráfica referentes a cada um dos três testes. Em seguida, na Figura 5-8, são apresentados os gráficos das séries de valores adquiridos em cada teste, ordenados por:

- tensão de *pull-in* do lado esquerdo do DUT;
- tensão de *pull-in* do lado direito do DUT;
- tensão de *pull-out* do lado esquerdo do DUT;
- tensão de *pull-out* do lado direito do DUT.

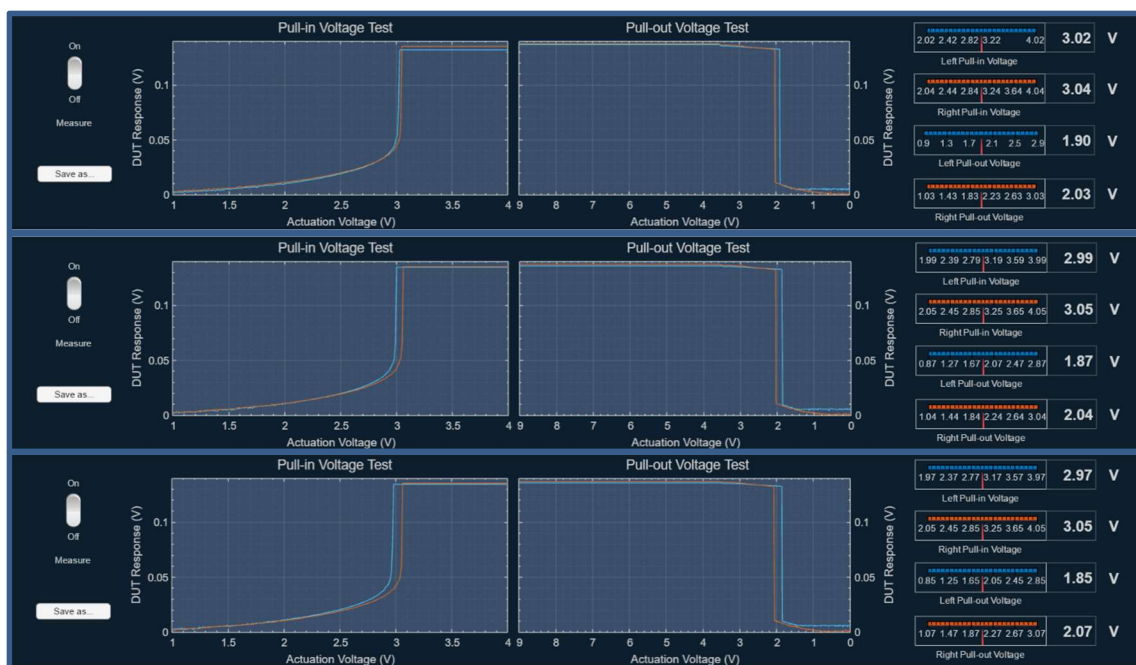


Figura 5-7 - Resultados da medição da tensão de *pull-in* e *pull-out* na GUI.

Os valores resultantes das medições efetuadas são apresentados nas capturas de ecrã da Figura 5-7 acima, sendo possível denotar que a variação máxima entre as três medições de cada parâmetro foi de 50 mV, devendo-se parte deste erro ao posicionamento da estrutura, sendo que estas tensões variam com a inclinação do DUT [24], e à resolução do algoritmo de medição utilizado, no qual é utilizada uma tensão de atuação com incrementos de 10 mV.

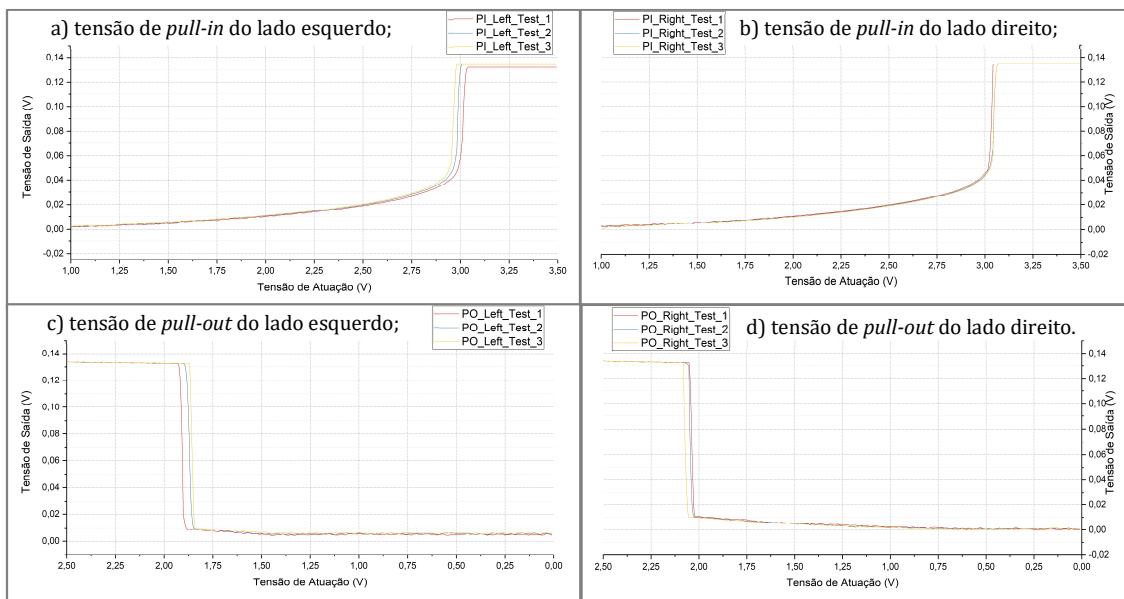


Figura 5-8 - Gráficos dos resultados da medição da tensão de pull-in e pull-out.

### 5.3.2. Frequency Sweep

De modo a comprovar o funcionamento da rotina de varrimento de frequências, descrita na secção 4.10, foram realizados três varrimentos a um acelerómetro MEMS capacitivo, com as seguintes configurações:

- Frequência inicial: 100 Hz;
- Frequência final: 1 kHz;
- Incremento de frequência: 10 Hz;
- Número de amostras por frequência: 1000;
- Amplitude do sinal aplicado: 1 V;
- *Offset* do sinal aplicado: 1 V.

## 5. Resultados

Através dos testes realizados, foi possível extrair a frequência de ressonância e fator de qualidade da estrutura, podendo estes resultados ser visualizados nas capturas de ecrã apresentadas na Figura 5-9.

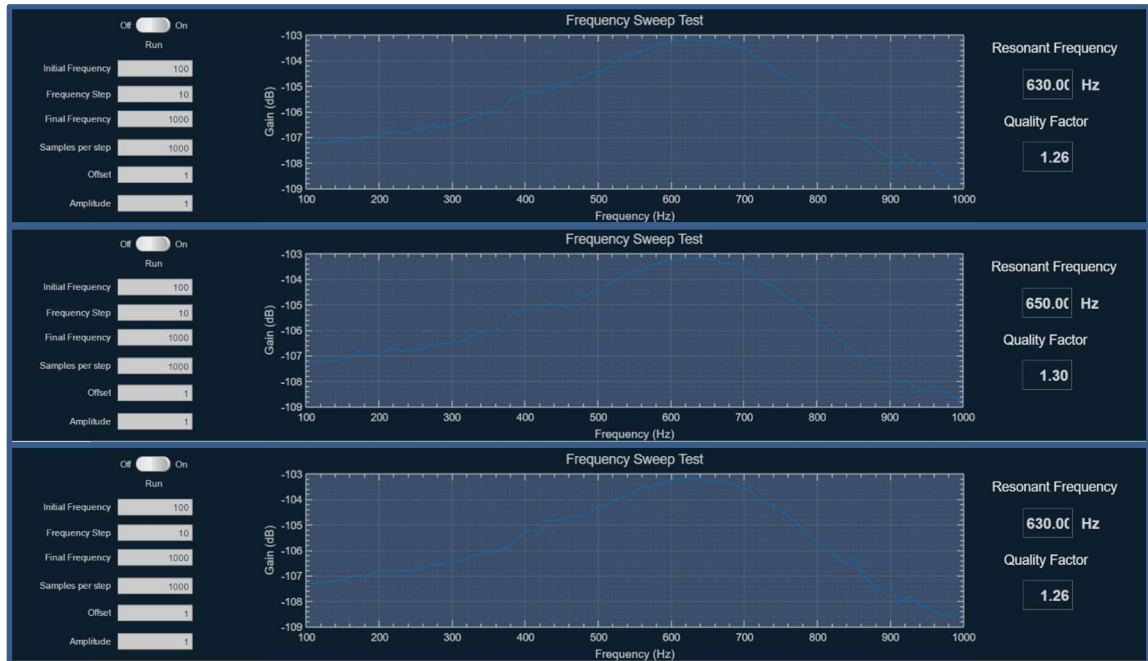


Figura 5-9 - Resultados do varrimento de frequências na GUI.

Na Figura 5-10, é possível observar um gráfico com as três séries de dados resultantes dos varrimentos efetuados de forma a tornar mais simples retirar ilações acerca da repetibilidade das medições obtidas com o algoritmo desenvolvido. Através deste gráfico, e em conjunto com a Figura 5-9, verifica-se que a variação máxima das medições da frequência de ressonância é de 20 Hz, correspondendo esta variação a duas vezes a resolução do teste efetuado, obtendo-se uma frequência de ressonância média de 636 Hz. Na medição do fator de qualidade da estrutura, foi possível obter um valor médio de fator de qualidade de 1,2867, sistema sub-amortecido, sendo a variação máxima entre testes de 0,04.

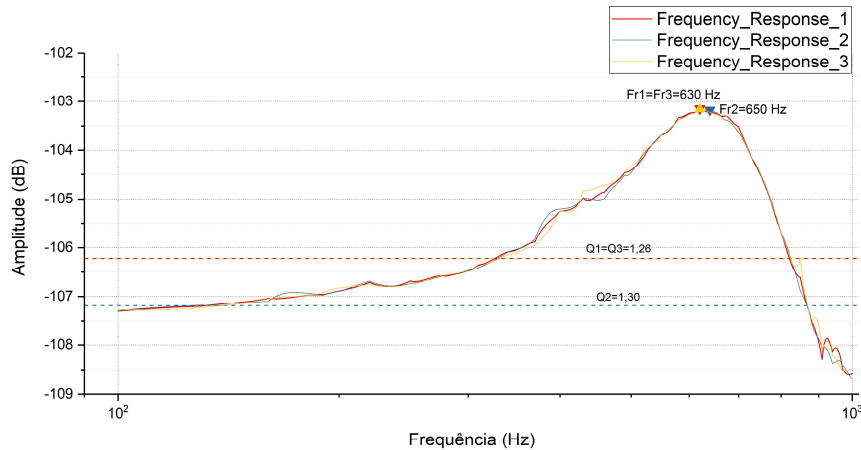


Figura 5-10 - Gráfico dos resultados do varrimento de frequências.

## 5.4. Conclusão dos Resultados

Neste capítulo, foram apresentados os resultados do desenvolvimento desta dissertação, exemplificando algumas das funcionalidades desta *toolbox*. Após análise dos resultados obtidos, é possível concluir que os mecanismos de configuração preenchem os requisitos perspetivados, garantindo a correta configuração de todos os parâmetros da plataforma de teste. Analogamente, os resultados dos modos de aquisição comprovam o funcionamento dos mecanismos de aquisição de dados. No capítulo seguinte, será apresentada uma conclusão acerca desta dissertação, assim como algumas sugestões para trabalho futuro.



# 6.

# Conclusões e Trabalho Futuro

Nesta dissertação, foi desenvolvida e apresentada uma *toolbox* com o intuito de expandir as funcionalidades da plataforma de teste de MEMS criada na Universidade do Minho em 2016. Esta *toolbox* consiste num sistema integrado composto por uma interface gráfica desenvolvida com recurso ao App Designer do MATLAB, um *driver* de comunicação desenvolvido utilizando o *software* Instrument Control Toolbox dos mesmos proprietários e vários blocos de *software* implementados ao nível do microprocessador ARM da plataforma de desenvolvimento Zedboard, que visam expandir as funcionalidades deste sistema de teste e caracterização de estruturas microeletromecânicas, como é o caso das rotinas de teste de *pull-in* e de varrimento de frequências.

Os resultados obtidos demonstram que o sistema foi desenvolvido com sucesso, garantindo que o utilizador da plataforma em questão tem ao seu dispor um sistema de teste mais completo e mais intuitivo de utilizar, com a opção de o fazer através da interface gráfica, através do *driver* de comunicação com recurso a um *script* de MATLAB ou diretamente através de um cliente TCP/IP, servindo-se do protocolo de comunicação criado.

Embora a rotina de varrimento de frequências tenha sido implementada, esta apenas extrai a amplitude da tensão de saída da estrutura sob teste, tornando de interesse a extração da fase do sinal, para garantir um mecanismo mais completo de aquisição da frequência de ressonância e fator de qualidade, e possibilitando a demonstração do diagrama de Bode.

É também relevante, para trabalho futuro, a migração das rotinas específicas de MEMS para *hardware* programável, na FPGA incluída na plataforma de teste, de forma a tornar os testes mais rápidos, assim como a adaptação do bloco de medição da tensão de *pull-in* e *pull-out*, de forma a adquirir também os tempos de *pull-in* e *pull-out* da estrutura em teste.

Outras melhorias à *toolbox* e conseqüentemente à plataforma de teste de MEMS incluem a criação de uma rotina de autoteste da plataforma, de modo a verificar o correto funcionamento de todos os componentes do sistema e um *trigger* para o *Live Mode* que garanta a visualização dos sinais adquiridos como um osciloscópio.



## Referências Bibliográficas

- [1] R. Jaeckel, "Production handling and testing of MEMS," *EE: Evaluation Engineering*, 2002. [Online]. Available: <http://www.evaluationengineering.com/production-handling-and-testing-of-mems>. [Accessed: 16-Dec-2016].
- [2] C. Ferreira, "Plataforma de Teste para MEMS," Universidade do Minho, 2016.
- [3] Yole Développement, "Why are some MEMS devices successful while others are not?," no. May, 2016.
- [4] Wang, *System-on-Chip Test Architectures: Nanometer Design for Testability*. 2008.
- [5] J. Korvink and O. Paul, *MEMS: A Practical Guide to Design, Analysis, and Applications*. 2006.
- [6] A. Introduction, M. January, and P. F. Partnership, *Prime Faraday Technology Watch An Introduction to MEMS ( Micro-electromechanical Systems ) MEMS has been identified as one of the most promising technologies for PRIME Faraday Partnership*, no. January. 2002.
- [7] N. Brito *et al.*, "Digital platform for wafer-level MEMS testing and characterization using electrical response," *Sensors (Switzerland)*, vol. 16, no. 9, 2016.
- [8] F. Beyeler, S. Muntwyler, and B. J. Nelson, "Wafer-level inspection system for the automated testing of comb drive based MEMS sensors and actuators," *2010 IEEE Int. Conf. Autom. Sci. Eng. CASE 2010*, pp. 698–703, 2010.
- [9] M. Vrânes, "High Volume MEMS Testing - Evolution, Challenges and the Future," *MEMS J.*, p. 26, 2016.
- [10] E. Lawrence, "Polytec Micro System Analyzer," 2012. [Online]. Available: <http://www.polytec.com/us/products/vibration-sensors/microscope-based-systems/msa-500-micro-system-analyzer/>. [Accessed: 20-Dec-2016].
- [11] FEMTO Tools, "FT-MPS02 MEMS Probestation." [Online]. Available: <http://www.femtotools.com/products/ft-mps02/description/>. [Accessed: 16-Dec-2016].

- [12] Instrumentation and technology for MEMS and Sensors, “ITMEMS MEMS Characterization Platform, Solutions for Microsystems Characterization.”
- [13] “Avnet Product Brief ZedBoard,.” [Online]. Available: [http://zedboard.org/sites/default/files/product\\_briefs/5066-PB-AES-Z7EV-7Z020-G-V1b.pdf](http://zedboard.org/sites/default/files/product_briefs/5066-PB-AES-Z7EV-7Z020-G-V1b.pdf). [Accessed: 14-Jan-2017].
- [14] S. C. Engine, A. A. User, T. Abaqus, and T. Star-ccm, “Target Applications,” pp. 5411–5448.
- [15] Xilinx, “Zynq-7000 All Programmable SoC Data Sheet: Overview,” *Ds190*, no. 11, p. 25, 2017.
- [16] Real Time Engineers Ltd., “The FreeRTOS™ Reference Manual,” 2016. [Online]. Available: <http://www.freertos.org/>. [Accessed: 18-Jan-2017].
- [17] “RS232 terminal, serial port terminal, udp terminal, tcp/ip client server - Hercules.” [Online]. Available: [http://www.hw-group.com/products/hercules/index\\_en.html](http://www.hw-group.com/products/hercules/index_en.html). [Accessed: 25-Feb-2017].
- [18] “lwIP - A Lightweight TCP/IP stack - Summary [Savannah].” [Online]. Available: <http://savannah.nongnu.org/projects/lwip/>. [Accessed: 12-Jan-2017].
- [19] A. Singh, G. Singh, S. Kavleen, K. Oberoi, and J. Kaur, “File Transfer Using Secure Sockets in Linux Environment.”
- [20] Matlab, “Instrument Control Toolbox Users Guide,” *MATLAB Man.*, 2009.
- [21] The MathWorks Inc, “MATLAB App Building,” p. 500, 2017.
- [22] The MathWorks Inc, “MATLAB App Building,” 2017. [Online]. Available: [https://www.mathworks.com/help/pdf\\_doc/matlab/buildgui.pdf](https://www.mathworks.com/help/pdf_doc/matlab/buildgui.pdf). [Accessed: 16-Feb-2017].
- [23] “Signal Processing Toolbox™ User’s Guide R2017b.”
- [24] V. Lima, J. Cabral, and L. Rocha, “Mecanismo de compensação de temperatura para sensores MEMS baseados na tensão de Pull-in.” p. 79, 2013.
- [25] I. To and A. Vibrometer, “Mems: a,” pp. 663–666, 2010.

# **ANEXOS**

# A: Mensagens de Comunicação

Mensagens Configuração						
Mensagem	ID		Valores	Unidade	Ação	
	Min	Max				
RC<ID>:ADC:FREQ:<VALOR>	1	2	0 a 17500000	Hz	Configura a frequência de amostragem de um canal<ID>	
RC<ID>:ADC:GAIN:<VALOR>	1	2	0, 1, 2, 3, 4, 5, 6	dB	Configura o ganho do ADC de um canal<ID>	
RC<ID>:AC_DC:<VALOR>	1	2	AC, DC	String	Seleciona a entrada do canal <ID> como AC ou DC	
RC<ID>:CD:<VALOR>	1	2	C, D	-	Seleciona a entrada do canal <ID> como Charge Amplifier ou Direto	
RC<ID>:SD:<VALOR>	1	2	S, D	-	Seleciona a entrada do canal <ID> como Single Ended ou Diferencial	
RC<ID>:CG:<VALOR>	1	2	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	-	Seleciona o ganho do canal de entrada <ID>	
RC<ID>:GAIN:<VALOR>	1	2	-6 a 26	-	Configura o ganho do ADC de um canal<ID>	
DAC<ID>:SEL:WG	1	4	-	-	Seleciona como saída do DAC<ID> o Wave Generator	
DAC<ID>:SEL:ADC	1	4	-	-	Seleciona como saída do DAC<ID> o ADC	
DAC<ID>:SEL:PHS	1	4	-	-	Seleciona como saída do DAC<ID> o fase	
DAC<ID>:SEL:AMP	1	4	-	-	Seleciona como saída do DAC<ID> o amplitude	
DAC<ID>:SEL:I	1	4	-	-	Seleciona como saída do DAC<ID> o I	
DAC<ID>:SEL:Q	1	4	-	-	Seleciona como saída do DAC<ID> o Q	
DAC<ID>:WG:TYPE:SIN	1	4	-	-	Seleciona o tipo de onda sinusoidal para o DAC<ID>	
DAC<ID>:WG:TYPE:TRI	1	4	-	-	Seleciona o tipo de onda triangular para o DAC<ID>	
DAC<ID>:WG:TYPE:SQR	1	4	-	-	Seleciona o tipo de onda quadrada para o DAC<ID>	
DAC<ID>:WG:TYPE:RAMP	1	4	-	-	Seleciona o tipo de onda rampa para o DAC<ID>	
DAC<ID>:WG:TYPE:PWM	1	4	-	-	Seleciona o tipo de onda PWM para o DAC<ID>	
DAC<ID>:WG:TYPE:DC	1	4	-	-	Seleciona DC para o DAC<ID>	
DAC<ID>:WG:FREQ:<VALOR>	1	4	1 a 7000000	Hz	Altera a frequência da onda do gerador <ID> para <valor> Hz	
DAC<ID>:WG:AMP:<VALOR>	1	4	-10 a 10	Volts	Altera a amplitude da onda do gerador <ID> para <valor> Volts	
DAC<ID>:WG:OFFSET:<VALOR>	1	4	-10 a 10	Volts	Altera o offset da onda do gerador <ID> para <valor> Volts	
DAC<ID>:WG:DUTY:<VALOR>	1	4	0 a 100	%	Altera o duty cycle da onda quadrada ou triangular do gerador <ID> para <valor> %	
DAC<ID>:WG:PHASE:<VALOR>	1	4	-180 a 180	Graus	Altera o desfasamento da onda do gerador <ID> para <valor> graus	
LI<ID>:DEC:<VALOR>	-	-	0 a 1000000	-	Fator de decimação do amplificador Lock-In	
LI:<id>:B2:<VALOR>	PRE	POST	-inf a inf	-	Coefficiente B2 do filtro	
LI:<id>:B1:<VALOR>	PRE	POST	-inf a inf	-	Coefficiente B1 do filtro	
LI:<id>:A2:<VALOR>	PRE	POST	-inf a inf	-	Coefficiente A2 do filtro	
LI:<id>:A1:<VALOR>	PRE	POST	-inf a inf	-	Coefficiente A1 do filtro	
LI:<id>:A0:<VALOR>	PRE	POST	-inf a inf	-	Coefficiente A0 do filtro	
LI:<id>:G:<VALOR>	PRE	POST	-inf a inf	-	Coefficiente G do filtro	
LI<ID>:FREQ:<VALOR>	-	-	1 a 5000000	Hz	Valor de frequência para geração de portadora do amplificador Lock-In	
LI<ID>:GAIN:<VALOR>	-	-	-6 a 26	dB	Valor de amplitude para a geração de portadora do amplificador Lock-In	
GET:AMP:<VALOR>	-	-	0 a 99000000	-	Adquire <valor> amostras da Amplitude	
GET:PHS:<VALOR>	-	-	0 a 99000000	-	Adquire <valor> amostras da Fase	
GET:I:<VALOR>	-	-	0 a 99000000	-	Adquire <valor> amostras de I	
GET:Q:<VALOR>	-	-	0 a 99000000	-	Adquire <valor> amostras de Q	
GET:ADC:<VALOR>	-	-	0 a 99000000	-	Adquire <valor> amostras de ADC	
GET:MULT:<VALOR>	-	-	0 a 99000000	-	Adquire <valor> amostras da Multiplicação	
GET:FILTER:<VALOR>	-	-	0 a 99000000	-	Adquire <valor> amostras após Filtro	
LIVE:AMP	-	-	-	-	Inicia a aquisição de dados da Amplitude em tempo real	
LIVE:PHS	-	-	-	-	Inicia a aquisição de dados da Fase em tempo real	
LIVE:I	-	-	-	-	Inicia a aquisição de dados de I em tempo real	
LIVE:Q	-	-	-	-	Inicia a aquisição de dados de Q em tempo real	
LIVE:ADC	-	-	-	-	Inicia a aquisição de dados do ADC em tempo real	
LIVE:FILTER	-	-	-	-	Inicia a aquisição de dados da Multiplicação em tempo real	
LIVE:MULT	-	-	-	-	Inicia a aquisição de dados após o Filtro em tempo real	
LIVE:STOP	-	-	-	-	Termina a aquisição de dados em tempo real	
SET:SFREQ:<VALOR>	-	-	1 a 17500000	Hz	Valor de frequência de amostragem	
SW:<ID>:<VALOR>	1	4	0, 1	-	Controlo do estado dos interruptores analógicos	
GET:PIV	-	-	-	-	Inicia a rotina de medição de pull-in e pull-out	
FSWEEP:<V1>:<V2>:<V3>:<V4>:<V5>:<V6>	-	-	V1: 1 a 7000000	Hz	Inicia a rotina de varrimento de frequências	
	-	-	V2: 1 a 7000000	Hz		
	-	-	V3: 1 a 7000000	Hz		
	-	-	V4: 1 a 99000000	-		
	-	-	V5: -10 a 10	Volts		
	-	-	V6: -10 a 10	Volts		

Figura A.1 - Mensagens de comunicação.