

# **Desenvolvimento de um codificador ótico sinusoidal e algoritmo para microcontrolador PIC32MZ**

**Nuno Ricardo Vaz Costa**

Dissertação para obtenção do Grau de Mestre em  
**Engenharia Eletromecânica**  
(2<sup>o</sup> ciclo de estudos ou mestrado integrado)

**Versão final após discussão**

Orientador: Prof. Doutor Pedro Dinis Gaspar  
Co-orientador: Prof. Doutor Eduardo Rodrigues

**Dezembro de 2023**



## Declaração de Integridade

Eu, Nuno Ricardo Vaz Costa, que abaixo assino, estudante com o número de inscrição M10236 de/o Mestrado em Engenharia Eletromecânica da Faculdade de Engenharia declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 06 /12 /2023

Assinado por: NUNO RICARDO VAZ COSTA  
Num. de Identificação: BI15010791  
Data: 2023.12.06 10.32.32 GMT Standard Time



(assinatura conforme Cartão de Cidadão ou preferencialmente  
assinatura digital no documento original se naquele mesmo formato)



# **Dedicatória**

Ao meu avô e avó, Manuel Martins Vaz e Carmina Pinto Gonçalves.



# Agradecimentos

Permitam-me a presunção de corrigir Álvaro de Campos: “a metafísica é também uma consequência de escrever a dissertação”.

Apesar de ter sido uma jornada com muitos mais obstáculos que os esperados consequentando no não alcance das metas definidas inicialmente, não deixou de ser um percurso de enorme aprendizagem quer a nível técnico quer a nível pessoal. Por isso mesmo, dedico os devidos agradecimentos na presente página a todas as pessoas e entidades que me acompanharam durante este percurso.

Em primeiro lugar, agradecer à Zonicstech e ao Eduardo pela disponibilização do espaço e equipamento de trabalho, mas principalmente por todo o conhecimento que me transmitiu durante esta jornada. Graças a esse conhecimento sou hoje um aspirante a engenheiro muito mais capaz de cumprir todas as tarefas e desafios que me são propostos a nível profissional.

Agradeço também ao departamento de Eng. Eletromecânica, em especial ao professor Pedro Gaspar, pela disponibilização de material que permitiu elaborar esta dissertação, mas acima de tudo pelos conhecimentos de engenharia que me foram transmitidos ao longo dos anos.

Quero também deixar uma grande palavra de apreço ao Sr. James Newton, que sem obrigação nenhuma, se disponibilizou a dar *feedback* sobre o trabalho desenvolvido e me ajudou a melhorar o modelo.

Um grande obrigado ao Martim por toda a disponibilidade e ajuda na impressão 3D e dicas dadas.

Agradecer à minha família por me ter possibilitado ter frequentado o curso de Eng. Eletromecânica e por todo o suporte ao longo dos anos.

A todos os meus amigos não só por terem feito disto um trajeto memorável mas também pela sua capacidade de transformar um momento menos bom num momento burlesco, apenas com o simples recurso à ironia e sarcasmo.

À Maia por todo o suporte, carinho, companheirismo e por ter sido uma grande parte deste caminho.



## Resumo

Na indústria há uma grande necessidade por sistemas cada vez mais eficientes e precisos. Exige-se às máquinas e braços robóticos trabalhos cada vez mais minuciosos e delicados. Isto provoca uma outra necessidade, métodos de controlo das mesmas mais exatos e conseqüentemente mais complexos. Métodos de controlo de posição dependem fortemente na aquisição de sinal, que traduzem a posição e velocidade da ferramenta da máquina ou garra do braço robótico. Neste aspeto, os codificadores têm grande relevância. Amplamente utilizados na indústria, estes dispositivos procuram traduzir movimentos mecânicos em sinais elétricos para posteriormente serem decodificados por algoritmos. O desenvolvimento de codificadores, apesar de não ser recente, é ainda objeto de estudo e de imensas pesquisas científicas com o objetivo de aumentar a sua resolução, exatidão e precisão. Dada a sua ampla utilização e vasto desenvolvimento, pode-se classificar os codificadores em ótico, magnético ou capacitivo quanto à fonte de sinal; incrementais ou absolutos quanto à referência de medição e lineares ou rotativos em conformidade com o tipo de medida.

Os codificadores óticos apesar de apresentarem algumas desvantagens face a outros, continuam a ser bastante utilizados por tenderem a permitir melhores valores de resolução. Desse modo, a presente dissertação propôs-se a desenvolver um codificador ótico sinusoidal baseado no projeto “Dexter 1” da empresa *Haddington Dynamics*. Desse mesmo projeto, foi aproveitada parcialmente a estrutura mecânica que contém peças essenciais ao codificador e permite conferir movimento ao mesmo através de um motor de passo. Foi também aproveitado integralmente o desenho eletrónico do mesmo projeto para a placa de circuito impresso do codificador. A conversão e processamento do sinal foi feito por um microcontrolador PIC32MZ numa placa de circuito impresso.

Foram desenvolvidos algoritmos de forma a configurar registos como temporizador, comparador de saída, conversão analógica-digital, etc. de forma a que o microcontrolador pudesse controlar o movimento do motor, adquirir e processar o sinal proveniente do codificador. Os resultados obtidos demonstram alguma margem para se aumentar a resolução do codificador e necessidade de uma melhor calibração dos sinais do codificador.

## Palavras-chave

Codificador ótico sinusoidal, Microcontrolador, Sistema de posicionamento, Impressão 3D, Braço robótico, Processamento de sinal



# **Abstract**

In the industry, there is a great demand for increasingly efficient and precise systems. Machines and robotic arms are expected to perform increasingly meticulous and delicate tasks. This creates another need for more accurate and consequently more complex control methods. Position control methods heavily rely on signal acquisition, which translates the position and velocity of the machine's tool or the robotic arm's gripper. In this aspect, encoders are highly relevant. Widely used in the industry, these devices aim to translate mechanical movements into electrical signals to be later decoded by algorithms. The development of encoders, although not recent, is still the subject of study and extensive scientific research with the aim of increasing their resolution, accuracy, and precision. Given their widespread use and extensive development, encoders can be classified as optical, magnetic, or capacitive depending on the signal source; incremental or absolute depending on the measurement reference; and linear or rotary according to the type of measurement.

Despite presenting some disadvantages compared to others, optical encoders are still widely used because they tend to allow for better resolution values. Thus, this dissertation aimed to develop a sinusoidal optical encoder based on the "Dexter 1" project by Haddington Dynamics. Partial use was made of the mechanical structure from the same project, which contains essential parts for the encoder and enables its movement through a stepper motor. The electronic design of the same project was also fully utilized for the encoder's printed circuit board. The signal conversion and processing were performed by a PIC32MZ microcontroller on a printed circuit board provided by Zonicstech.

Algorithms were developed to configure registers such as timer, output compare, analog-to-digital conversion, etc., so that the microcontroller could control the motor's movement, acquire and process the signal from the encoder. The obtained results demonstrate some room for increasing the encoder's resolution and the need for better calibration of the encoder's signals.

# **Keywords**

Sinusoidal optical encoder, Microcontroller, Positioning system, 3D printing, Robotic arm, Signal processing



# Índice

Dedicatória.....	iii
Agradecimentos .....	v
Resumo .....	vii
Abstract.....	ix
Índice .....	xi
Lista de Figuras.....	xv
Lista de Tabelas .....	xix
Nomenclatura .....	xxi
1. Introdução .....	1
1.1. Enquadramento .....	1
1.2. Objetivos e contribuição da dissertação .....	3
1.3. Visão geral e organização da dissertação.....	4
2. Estado da Arte .....	5
2.1. Codificadores Sinusoidais.....	5
2.1.1. Aquisição de Sinal .....	8
2.1.2. Conversão Analógico-Digital.....	10
2.1.3. Ruído e Análise de Fourier.....	12
2.1.4. Filtros Analógicos.....	14
2.1.5. Filtros Digitais.....	17
3. Materiais e Métodos .....	19
3.1. Microcontrolador PIC32MZ .....	19
3.1.1.1. Rotina de Serviço de Interrupção (RSI) .....	20
3.1.1.2. Oscilador.....	21
3.1.1.3. Portas de Entrada e Saída .....	22
3.1.1.4. Temporizador .....	23
3.1.1.5. Comparador de saída.....	25
3.1.1.6. Recetor-Transmissor Assíncrono Universal .....	27
3.1.1.7. Conversão Analógico-Digital.....	28

3.1.1.7.1.	ADCCON1.....	30
3.1.1.7.2.	ADCCON2 .....	31
3.1.1.7.3.	ADCCON3 .....	31
3.1.1.7.4.	ADCxTIME.....	32
3.1.1.7.5.	ADCIMCONy.....	33
3.1.1.7.6.	ADCTRGy.....	33
3.1.1.7.7.	ADCGIRQENy.....	34
3.1.1.7.8.	ADCANCON .....	34
3.1.1.7.9.	ADCDATAx .....	34
3.1.1.7.10.	ADCxCFG .....	35
3.1.1.8.	Rotina de Serviço de Interrupção .....	35
3.1.2.	Placa de circuito impresso Zonicstech.....	35
3.1.3.	Motor de passo .....	37
3.1.4.	<i>Driver</i> do Motor .....	38
3.1.4.1.	LED Infravermelho .....	40
3.1.4.2.	Fototransistor .....	41
3.1.5.	Pickit 4.....	41
3.1.6.	Conversor USB .....	42
3.1.7.	Impressão 3D.....	43
3.1.7.1.	Artillery Sidewinder X1.....	44
3.1.8.	Soldadura Manual .....	44
3.1.9.	Matlab.....	45
3.1.10.	Mplab X IDE.....	46
3.1.11.	SerialPlot .....	46
3.1.12.	Kicad.....	47
3.1.13.	Solidworks .....	48
4.	Protótipo .....	49
4.1.	Desenvolvimento do codificador ótico.....	49

4.1.1.	Desenho eletrônico.....	49
4.1.2.	Componente mecânica.....	54
4.1.3.	Montagem da estrutura.....	56
4.1.4.	Conexões .....	65
4.1.5.	Filtro Analógico.....	68
5.	Análise e Discussão de Resultados .....	71
5.1.	Primeiro Teste.....	71
5.1.1.	Preparação do primeiro teste.....	71
5.1.1.1.	Algoritmo das funções .....	72
5.1.1.1.1.	Função fontes de relógio.....	72
5.1.1.1.2.	Funções Timer .....	73
5.1.1.1.3.	Funções do comparador de saída.....	77
5.1.1.1.4.	Funcionamento das funções do conversor analógico digital .....	78
5.1.1.1.5.	Funções UART.....	81
5.1.1.1.6.	Configuração do Driver do motor de passo.....	85
5.1.1.1.7.	Fluxograma.....	85
5.1.2.	Escolha do Disco .....	88
5.1.3.	Resultados do primeiro teste .....	90
5.1.4.	Resolução de problemas do 1º teste.....	93
5.1.4.1.	Centrar sinais no eixo x .....	93
5.1.4.2.	Filtro Digital .....	97
5.1.4.3.	Simulação dos filtros em Matlab.....	100
5.1.4.4.	Aplicação do algoritmo do ângulo .....	103
5.1.5.	Recapitulação dos resultados do primeiro teste .....	107
5.2.	Segundo Teste.....	109
5.2.1.	Funções Filtro .....	109
5.2.2.	Função ângulo.....	111
5.2.3.	Funcionamento do 2º teste .....	117

5.2.4. Resultados .....	118
6. Conclusões.....	125
6.1. Conclusões Gerais .....	125
6.2. Sugestões de Trabalho Futuro.....	127
Referências Bibliográficas .....	129
Anexos .....	135

# Lista de Figuras

Figura 1 - Quadratura de dois sinais quadrados desfasados $90^\circ$ , adaptado de(TI, 2019). .....	6
Figura 2 - Função arco tangente aplicada aos sinais seno e cosseno. ....	7
Figura 3 - Função derivada do arco tangente com amplitude de dois PI radianos, aplicada aos sinais seno e cosseno. ....	7
Figura 4 - Intensidade de corrente num díodo ideal com tensão de joelho de 0,8V, adaptado de (Malvino & Bates, 2016).....	9
Figura 5 - Circuito divisor de tensão adaptado de (Robbins & Miller, 2012).....	10
Figura 6 - Circuito de amostragem e espera adaptado de(Kester, 2005).....	11
Figura 7 – Efeitos de uma função janela num sinal. Adaptado de (Kester & Analog Devices, 2003). ....	14
Figura 8 - Filtro RC (Adaptado de Robbins & Miller, 2013). ....	15
Figura 9 – Resposta em frequência (Adaptado de Robbins & Miller, 2013).....	17
Figura 10 – Resposta de fase (Adaptado de Robbins & Miller, 2013).....	17
Figura 11 - Exemplo de registo e correspondentes subregistos num PIC32MZ. ....	20
Figura 12. Funcionamento do pino de saída em Modo de Comparação Dupla com pulso contínuo e respetiva configuração. Adaptado de (Microchip Technology Inc., 2011). .....	26
Figura 13. Representação dos períodos de gatilho, conversão e retenção/conversão....	30
Figura 14 - Placa de Circuito Impresso Zonicstech. ....	36
Figura 13 - Motor de passo 42BYGMH809 (Hentai Motor, n.d.) ....	37
Figura 16. Diagrama Funcional do Driver A4988 (Allegro Microsystems, 2022) .....	39
Figura 17. Diagrama de Entradas e saídas do Driver A4988 (Allegro Microsystems, 2022) .....	39
Figura 18. Placa de circuito impresso do Driver A4988 (PTRobotics, n.d.) .....	40
Figura 19. Led Infravermelho LTE-4206 (LITE-ON Technology Corporation, n.d.-a) .40	
Figura 20. Foto transistor LTR-4206E (LITE-ON Technology Corporation, n.d.-b) ....	41
Figura 21. Relação entre corrente no coletor e irradiância na base adaptado de (LITE-ON Technology Corporation, n.d.-b) .....	41
Figura 22 – Pickit 4.....	42
Figura 23 - Conversor USB-Nano-485 .....	43
Figura 24 – Impressora 3D <i>Artillery Slidewinder X1</i> (3D Printer Universe, n.d.).....	44
Figura 25 - Estação de Soldadura com ferro e secador. ....	45
Figura 26 – Interface do <i>Mplab X IDE</i> (Microchip Technology Inc, 2022). ....	46

Figura 27 – Interface do SerialPlot. ....	47
Figura 28. Esquema elétrico dos circuitos led – fototransistor.....	50
Figura 29. Desenho elétrico do terminal que fornece a tensão de alimentação e recolhe a tensão à saída do fototransistor. ....	51
Figura 30 - Gráfico de impedância, resistência e reactância da <i>ferrite bead</i> em função da frequência. Adaptado de (Yuden, n.d.) .....	52
Figura 31. Desenho da placa de circuito impresso .....	53
Figura 32 - Circuito impresso já com componentes por soldar .....	53
Figura 33 - Circuito impresso já com componentes soldados. ....	53
Figura 34 - Várias vistas do Interruptor de Radiância.....	54
Figura 35 - Vista superior dos furos do Interruptor de Radiância. ....	54
Figura 36 – Disco Codificador .....	55
Figura 37 – Cobertura do Codificador .....	56
Figura 38 - Motor com os suportes inferior e superior.....	57
Figura 39 - Coluna da base com um rolamento. ....	58
Figura 40 – Coluna da base com um rolamento e motor encaixados.....	58
Figura 41 – Coluna da base com ambos os rolamentos e motor encaixados.....	58
Figura 42 – Base de montagem superior e coluna da base com o motor. ....	59
Figura 43 – Aperto dos parafusos da base de montagem superior. ....	59
Figura 44 – Base de Montagem inferior com as 6 barras de carbono encaixadas. ....	60
Figura 45 – Base de Montagem inferior já montada na estrutura.....	60
Figura 46 - Disco Codificador na estrutura.....	61
Figura 47 – Estrutura com duas barras de carbono nos suportes do motor. ....	62
Figura 48 – Pivô já encaixado na estrutura. ....	62
Figura 49 – Interruptor de radiância.....	63
Figura 50 – Interruptor de radiância na estrutura. ....	63
Figura 51 - Cobertura do codificador apertada à estrutura.....	64
Figura 52 - Placa de circuito impresso montada na estrutura. ....	64
Figura 53. Ligações entre os diferentes dispositivos.....	66
Figura 54 – Estrutura montada com a placa da Zonicstech e <i>breadboard</i> . ....	67
Figura 55 – Diagrama de bode do Filtro da Condição A.....	70
Figura 56 - Diagrama de bode do Filtro da Condição B.....	70
Figura 57 - Fluxograma do funcionamento da função de configuração dos <i>timers</i> . ....	74
Figura 58 - Escolha do Préescalar para o <i>Timer1</i> . ....	75
Figura 59 - Escolha do Préescalar do <i>Timer2</i> ao <i>Timer9</i> . ....	76

Figura 60 - Fluxograma do funcionamento da função de configuração do comparador de saída 3. ....	78
Figura 61 - Fluxograma da função de configuração dos canais do ADC. ....	80
Figura 62 - Fluxograma da função de módulo ADC. ....	81
Figura 63 - Fluxograma da função de configuração da UART2. ....	83
Figura 64 - Fluxograma da função encarregue de enviar dados por comunicação UART. ....	84
Figura 65 - Fluxograma geral do funcionamento do 1º teste. ....	87
Figura 66 - Fluxograma da máquina de estados do 1º teste.....	88
Figura 67 – Sinal sinusoidal de 360º do 1º disco.....	89
Figura 68 - Sinal sinusoidal de 360º do disco utilizado na dissertação.....	89
Figura 69 – Amostra de 100mil conversões de cada sinal do 1º teste.....	91
Figura 70 – Gráfico da magnitude de frequências da condição A.....	92
Figura 71 - Gráfico da magnitude de baixas frequências da condição A. ....	92
Figura 72 – Comparação da magnitude das frequências entre a condição A e B.....	93
Figura 73 - Número de ciclos que o sinal com o filtro com o coeficiente 0,9999.....	94
Figura 74 - Número de ciclos que o sinal com o filtro com o coeficiente 0,99999.....	95
Figura 75 - Histograma de valores médios por ciclo elétrico quando subtraído o valor médio. ....	96
Figura 76 - Histograma de valores médios por ciclo elétrico quando aplicado o filtro digital de passo a alto.....	97
Figura 77 - Intervalo de tempo entre o cálculo do ângulo e o próximo ciclo de conversão. O sinal a amarelo corresponde à interrupção para conversão e o sinal verde corresponde á função de cálculo do ângulo.....	99
Figura 78 - Tempo de processamento do algoritmo do filtro IIR para 5 coeficientes...	99
Figura 79 - Tempo de processamento do algoritmo do filtro IIR para 6 coeficientes..	100
Figura 80 - Tempo de processamento do algoritmo do filtro IIR para 7 coeficientes.	100
Figura 81 – Comparação da magnitude de frequências entre dados filtrados e não filtrados da condição A. ....	101
Figura 82 - Comparação da magnitude de frequências entre dados filtrados e não filtrados da condição B. ....	101
Figura 83 - Comparação da magnitude de frequências dos dados filtrados entre condição A e condição B.....	102
Figura 84 – Comparação das primeiras conversões entre dados filtrados e por filtrar no domínio do tempo.....	103
Figura 85 – Gráfico do ângulo elétrico.....	104

Figura 86 - Representação cartesiana de 3 ciclos elétricos dos sinais sinusoidais da Figura 85.....	104
Figura 87 - Representação dos sinais da Figura 85. ....	105
Figura 88 – Gráfico do ângulo elétrico. ....	105
Figura 89 - Representação cartesiana do sinal da Figura 88.....	106
Figura 90 - Representação dos sinais da Figura 88.....	106
Figura 91 - Histograma da relação entre amplitudes de cada sinal por ciclo. ....	108
Figura 92 - Fluxograma da função “ <i>Filter_IIR</i> ”. ....	110
Figura 93 – Fluxograma do algoritmo do cálculo do ângulo. ....	114
Figura 94 – Tempo de processamento da função <i>angle_calc</i> utilizando atan2 para o cálculo do ângulo elétrico.....	115
Figura 95 - Tempo de processamento da função <i>angle_calc</i> utilizando nova fórmula. ....	115
Figura 96 – Diferença entre cálculo pela fórmula atan2 e nova fórmula. ....	116
Figura 97 – Histograma da diferença entre cálculo pela fórmula atan2 e nova fórmula. ....	116
Figura 98 – Funcionamento do 2º teste. ....	118
Figura 99 – Magnitude de frequências da condição A sinal 1.....	119
Figura 100 - Magnitude de frequências da condição A sinal 2. ....	119
Figura 101 - Magnitude de frequências da condição B Sinal 1.....	120
Figura 102 - Magnitude de frequências da condição B Sinal 2.....	120
Figura 103 – Ângulo elétrico da condição A. ....	121
Figura 104 - Ângulo elétrico da condição B. ....	122
Figura 105 - Ângulo mecânico da condição A. ....	123
Figura 106 - Ângulo mecânico da condição B. ....	123
Figura 107 – Comparação da magnitude das frequências entre a condição A e a condição B. ....	124

## Lista de Tabelas

Tabela 1. Função predefinida e recursos disponíveis para cada pin apresentado. (Nota: todos os pinos estão disponíveis para serem utilizados como entradas e/ou saídas assim como têm outros recursos associados que não foram mencionados na presente dissertação) .....	37
Tabela 2. Configuração da resolução de passo. “L” – Nível Baixo de Tensão; “H” – Nível Alto de Tensão (Allegro Microsystems, 2022).....	40
Tabela 3 – Peças Impressas em 3D .....	56
Tabela 4 - Materiais para a construção da estrutura .....	57
Tabela 5. Ligação entre pinos do microcontrolador e outros dispositivos. ....	67
Tabela A1. Relação entre as fontes de relógio e atributos do microcontrolador. Adaptado de (Microchip Technology Inc., 2013b).....	135
Tabela A2. Registo PBxDIV. Adaptado de (Microchip Technology Inc., 2021) .....	135
Tabela A3. Registo T1CON. Adaptado de (Microchip Technology Inc., 2021).....	136
Tabela A4. Registo TxCON. Adaptado de (Microchip Technology Inc., 2021) .....	137
Tabela A5. Vetores e Registos das rotinas de serviço e interrupção de cada Timer. Adaptado de (Microchip Technology Inc., 2021) .....	138
Tabela A6. Registos das rotinas de serviço e interrupção de cada Timer. Adaptado de (Microchip Technology Inc., 2021).....	138
Tabela A7. Relação entre os módulos do Comparador de saída e os Módulos Timer. Adaptado de (Microchip Technology Inc., 2021) .....	139
Tabela A8. Registo OCxCON. Adaptado de (Microchip Technology Inc., 2021) .....	140
Tabela A9. Registo OCxCON. Adaptado de (Microchip Technology Inc., 2021) .....	141
Tabela A10. Registos das rotinas de serviço e interrupção de cada comparador de saída. Adaptado de (Microchip Technology Inc., 2021) .....	141
Tabela A11. Registo UxMODE. Adpatado de (Microchip Technology Inc., 2021) .....	142
Tabela A12. Registo UxSTA da posição 6 à posição 31. Adaptado de (Microchip Technology Inc., 2021) .....	143
Tabela A13. Registo UxSTA da posição 6 à posição 31. Adaptado de (Microchip Technology Inc., 2021) .....	144
Tabela A14. Vetores e Registos das rotinas de serviço e interrupção de cada UART. Adpatado de (Microchip Technology Inc., 2021) .....	144
Tabela A15. Registos das rotinas de serviço e interrupção de cada UART. Adpatado de (Microchip Technology Inc., 2021).....	145

Tabela A16. Registo ADCCON1. Adpatado de (Microchip Technology Inc., n.d.).....	145
Tabela A17. Registo ADCCON2. Adpatado de(Microchip Technology Inc., 2021).....	146
Tabela A18. Registo ADCCON3. Adaptado de (Microchip Technology Inc., 2021).....	147
Tabela A19. Registo ADCCON3. Adaptado de (Microchip Technology Inc., 2021).....	148
Tabela A20. Módulos ADC e respetivas entradas analógicas. Adaptado de (Microchip Technology Inc., 2021) .....	149
Tabela A21. Valores e funções dos subregistos TRGSRCx. Adaptado de (Microchip Technology Inc., 2021) .....	149
Tabela A22. Registo ADCTRGMODE desde a posição 16 à posição 31. Adaptado de(Microchip Technology Inc., 2021) .....	150
Tabela A23. Subregisto WKUPCLKCNT. Adaptado de (Microchip Technology Inc., 2021) .....	151
Tabela A24. Vetor e Registos das rotina de serviço e interrupção global do ADC. Adpatado de (Microchip Technology Inc., 2021) .....	151

# Nomenclatura

## Geral:

a	coeficientes de retroalimentação;
b	coeficientes de alimentação direta;
C	Capacidade [F];
f	frequência [Hz];
I	Intensidade de Corrente [A];
j	Unidade imaginária;
L	Indutância [H];
N	Número de furos;
R	Resistência [ $\Omega$ ];
Res	Resolução;
t	Tempo [s];
V	Tensão [V];
X	Reactância [ $\Omega$ ];
Z	Impedância [ $\Omega$ ];

## Índices inferiores:

c	Frequência de corte;
conv	conversão;
in	Entrada;
out	Saída;
samp	amostragem;

## Simbologia grega:

$\theta$	ângulo mecânico [ $^{\circ}$ ];
$\alpha$	ângulo elétrico [ $^{\circ}$ ];
$\omega$	frequência angular [rad/s];

## ***Acrónimos:***

3D	Tridimensional;
ADC	<i>Analog-to-Digital Converter;</i>
CAD	<i>Computer Aided Manufacturing;</i>
CAN	<i>Controller Area Network;</i>
DAC	<i>Digital-to-Analog Converter;</i>
DMA	<i>Direct Memory Access;</i>
I2C	<i>Inter-Integrated Circuit;</i>
IFR	<i>International Federation of Robotics;</i>
SPI	<i>Serial Peripheral Interface;</i>
PCB	<i>Printed Circuit Board;</i>
PWM	<i>Pulse Width Modulation</i>
RAM	<i>Random Access Memory;</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>

# 1. Introdução

## 1.1. Enquadramento

Em outubro do ano transato, a Federação Internacional de Robótica (IFR -*International Federation of Robotics*) reportou que foram instalados 517385 robôs industriais em 2021. Na altura, 3,5 milhões de robôs estavam instalados em fábricas de todo o mundo, sendo o número mais alto de sempre e com tendência a aumentar, tendo em conta o crescimento exponencial nestes últimos anos (International Federation of Robotics, 2022).

Para além do crescimento em ambientes industriais, os robôs têm vindo cada vez mais a substituir máquinas industriais dedicadas, graças às vantagens que apresentam como a sua flexibilidade, espaço de trabalho alargado, graus de liberdade e custo-eficiência. No entanto, estes ainda apresentam uma grande desvantagem no que toca à exatidão. Por isso mesmo, têm sido desenvolvidos imensos estudos de forma a colmatar esta inconveniência (Kim *et al.*, 2019).

De forma a ultrapassar este obstáculo, diversos métodos de calibração têm sido desenvolvidos (Chen & Dong, 2013). Um dos métodos desenvolvidos passa por identificar os parâmetros cinemáticos iterativamente, através do modelo cinemático de parâmetros Denavit-Hartenberg e de um rastreador laser. Isto permite corrigir alguns erros de medida dos elos do respetivo robô (Ye *et al.*, 2006). Uma vez que quando o robô está sobre carga os elos podem sofrer flexão e isso provocará erro de posicionamento da garra, têm sido também desenvolvidos métodos de forma a poder-se determinar essa flexão em cada elo. De maneira a poder conhecer-se desvios ao trajeto esperado, foram desenvolvidos métodos utilizando dispositivos externos como câmaras ou rastreadores laser na garra e codificadores secundários nas juntas, permitindo criar um sistema de controlo com *feedback*, de modo a possibilitar a compensação do erro (Kim *et al.*, 2019).

No entanto, estes métodos referidos têm apenas como objetivo corrigir erros ou tentar colmatar limitações dos sistemas de posicionamento. Sistemas de posicionamento não são mais do que métodos de medição que, por sua vez, se definem como a quantificação do tamanho, quantidade ou grau em unidades específicas (Gao, 2010). O desenvolvimento constante da tecnologia veio permitir que novos, mais precisos e exatos métodos e dispositivos de medição fossem desenvolvidos (Yu *et al.*, 2021a). Estes dispositivos são sempre acompanhados de um transdutor que têm como propósito

traduzir características físicas de um sistema em sinais elétricos (que podem ser corrente, tensão ou carga) (Fraden, 2004). Dada a necessidade de robôs industriais mais precisos e exatos, a capacidade dos sensores de posicionamento ganha elevada relevância. Características como a resolução e ruído do sinal dos sensores devem ser tidas em conta. Alguns dos transdutores mais utilizados em sistemas de posicionamento são sensores resistivos, sensores capacitivos, sensores eletromagnéticos e sensores óticos (Fleming, 2013).

Os codificadores, um dos sistemas de posicionamento mais utilizados na robótica, fazem uso de alguns destes sensores. A classificação destes sistemas de posicionamento pode ser feita em conformidade com o tipo de sensor utilizado, embora possam também ser diferenciados em codificadores lineares ou rotacionais, independentemente da fonte do sinal de medida.

Geralmente, os codificadores de alta resolução são desenhados de forma a terem como saída dois sinais sinusoidais em quadratura, embora sejam utilizadas outras formas de sinal. Sinais deste tipo tendem a permitir uma maior resolução de posicionamento (Hoang & Jeon, 2011). No caso de codificadores rotativos, o ângulo pode ser retirado através do arcotangente destes dois sinais (Hao *et al.*, 2009).

Os codificadores magnéticos têm vindo a ser cada vez mais utilizados em ambientes industriais, uma vez que os sinais produzidos pelos mesmos não são tão sensíveis à sujidade, poeira, temperatura e vibrações (Hoang & Jeon, 2011). Para este tipo de codificadores são frequentemente utilizados sensores de efeito de Hall. De forma a formarem-se ondas sinusoidais desfasadas  $90^\circ$ , é colocado um sensor a  $90^\circ$  do outro. No centro do círculo estará um íman que rodará solidário com o veio. A rotação do íman fará com que os sensores formem ondas sinusoidais desfasadas  $90^\circ$  em que cada ciclo corresponderá a uma revolução do veio. De forma a aumentar-se a resolução são frequentemente utilizados múltiplos sensores distanciados igualmente (Hao *et al.*, 2009).

De forma análoga, através do princípio da capacidade dos condensadores, foram desenvolvidos codificadores capacitivos. Estes possuem dois pratos dispostos paralelamente a uma certa distância, um fixo e um que terá movimento circular. O princípio passa por um circuito transmissor do prato estático, que é uma placa de circuito impresso, provocando uma diferença potencial variável no circuito da placa rotativa, que por sua vez “refletirá” essa diferença potencial ao circuito recetor da placa estática, que

deverá estar ligado a um ADC (conversor analógico-digital). Este tipo de codificadores apresenta vantagens semelhantes aos condensadores magnéticos no que toca ao funcionamento em ambientes fabris (Zheng *et al.*, 2015). Apesar das vantagens que apresentam codificadores como o codificador magnético e capacitivo, os codificadores óticos continuam a ser muito utilizados devido à elevada resolução que se pode obter. De forma geral, os codificadores sinusoidais deste tipo utilizam uma fonte luminosa e um sensor ótico. Entre os dois estará uma estrutura que com o movimento do motor provocará radiação oscilante no sensor ótico. A resolução deste codificador dependerá do movimento do motor necessário de forma a haver um ciclo sinusoidal no sensor (Watanabe *et al.*, 2014; Yu *et al.*, 2021b).

Embora a geração do sinal constitua uma parte importantíssima, o trabalho dos codificadores não fica por aqui. É necessário converter-se o sinal analógico em digital onde a resolução e frequência do ADC contribuirão para a resolução de todo o sistema. Fatores como a média do sinal, diferença de amplitudes, deformações no sinal e ruído poderão provocar problemas no que toca à exatidão do posicionamento e por isso é sempre necessária alguma calibração de forma a corrigirem-se estes possíveis desvios (Hoang & Jeon, 2007, 2011).

## **1.2. Objetivos e contribuição da dissertação**

Nos dias que correm, praticamente todos os sistemas de robótica e maquinaria industrial necessitam de sistemas de controlo de alta resolução e exatidão. Por sua vez, os sistemas de controlo necessitam de sistemas de posicionamento. Neste aspeto, os codificadores ganham grande relevância pois são amplamente utilizados na indústria de forma a fornecer informações precisas sobre a posição, velocidade e direção de movimento de sistemas mecânicos em tempo real.

Os codificadores, de modo geral, traduzem os movimentos mecânicos em sinais elétricos para posteriormente serem processados por microcontroladores. Quanto maior a resolução e exatidão do sinal extraído, menor será a discrepância entre a posição desejada e a posição alcançada pelo manipulador robótico ou ferramenta da máquina. Dada a crescente necessidade de sistemas cada vez mais precisos e eficientes, os codificadores tornam-se indispensáveis nos mesmos e exige-se aos mesmos uma constante evolução.

Um dos codificadores mais utilizados é o codificador ótico, graças à sua capacidade de obter sinais de elevada resolução. Por isso mesmo, esta dissertação propõe-se a desenvolver um codificador ótico sinusoidal baseado no projeto “Dexter 1” da empresa *Haddington Dynamics*.

O projeto do “Dexter 1” está parcialmente disponibilizado em *opensource*. Esta dissertação fez uso de parte da estrutura mecânica disponibilizada em modelos tridimensionais, usando integralmente modelos para algumas peças, adaptando algumas às necessidades e desafios da dissertação e projetar e desenhar do início outras. O sistema eletrónico do codificador ótico também está disponibilizado em *opensource*, de onde esta dissertação fez uso integral do mesmo.

### **1.3. Visão geral e organização da dissertação**

De seguida segue-se uma breve descrição dos seis capítulos da presente dissertação.

O primeiro capítulo, denominado Introdução, são descritos de forma sucinta os tipos de codificadores e a sua relevância em sistemas de posicionamento e controlo.

O segundo capítulo começa por abordar o funcionamento de codificadores óticos e é feita uma explicação dos métodos de aquisição e processamento de sinal utilizados na presente dissertação.

No terceiro capítulo são referidos todos os materiais utilizados, explicando o seu funcionamento e propósito na presente dissertação. São também abordados todos os *softwares* utilizados na mesma.

O quarto capítulo pretende fazer apresentação do protótipo, demonstrar os componentes mecânicos e respetiva montagem, expor a vertente eletrónica e o seu funcionamento e apresentar as respetivas conexões entre as duas vertentes.

O quinto capítulo objetiva demonstrar os testes feitos ao codificador e apresentar e explicar os algoritmos para os mesmos. Neste capítulo são ainda abordadas algumas das limitações do respetivo protótipo.

O último capítulo apresenta as limitações e deficiências do modelo desenvolvido e refere alguns trabalhos futuros que podem vir a colmatar essas deficiências

## 2. Estado da Arte

### 2.1. Codificadores Sinusoidais

Em diversas áreas, nomeadamente na robótica é essencial que se consiga retirar informações sobre a rotação do motor como posicionamento, velocidade e direção (Colao, 2020). De forma a obter-se esses dados são utilizados codificadores, existindo variados tipos. Um dos tipos de codificador mais utilizado é o codificador ótico. Esta gama de codificadores caracteriza-se pela utilização de fontes luminosas e sensores óticos. Estes estão separados por um disco, solidário com o veio do motor, com secções opacas e transparentes no mesmo raio da fonte de iluminação e sensor ótico (Kester, 2009b). A rotação do disco, dependendo do formato das secções, fará com que a saída do sensor ótico seja uma onda quadrada, ou uma onda sinusoidal que depois pode ser convertida para uma onda quadrada de sinais altos e baixos de tensão através de um comparador (Texas Instruments Incorporated, 2019). Isto fará com que a resolução do posicionamento seja 360 a dividir pelo número de secções transparentes e opacas. Ou seja, se o disco tiver 100 secções opacas e transparentes poder-se-á saber o posicionamento do veio de  $1,8^\circ$  em  $1,8^\circ$ . Este funcionamento refere-se aos codificadores óticos incrementais, que têm a limitação de não transmitirem informação sobre a direção do movimento e da medição não ser absoluta. Para contornar estas limitações por vezes são adicionados mais dois sensores óticos e mais duas fontes de iluminação, de forma a retirar-se esta informação. Já os codificadores óticos absolutos são geralmente constituídos por  $n$  sensores óticos e  $n$  fontes de iluminação, sendo  $n$  o número de bits da palavra que traduzirá a posição. Cada sensor estará alinhado com uma fonte de iluminação, mas cada par estará disposto num raio diferente do disco. O disco deste tipo de codificadores tem secções transparentes e opacas como nos codificadores incrementais, mas cada par sensor/fonte de iluminação estará sujeito a secções de comprimentos angular diferentes permitindo assim que à saída dos sensores se forme uma palavra de  $n$  bits. (Kester, 2009b).

Há ainda codificadores incrementais que utilizam dois sensores óticos e duas fontes de iluminação desfasados  $90^\circ$  elétricos. Isto é, duas secções transparentes consecutivas perfazem  $360^\circ$  elétricos. Se estas estiverem separadas por  $3,6^\circ$  físicos, quer dizer que o par fonte de iluminação-sensor estará separado por  $0,9^\circ$  físicos. Depois de os sinais serem convertidos, a ondas quadradas pelo comparador ter-se-á dois sinais quadrados desfasados  $90^\circ$  o que permitirá aumentar a resolução por ciclo elétrico como

demonstrado na Figura 1 (TI, 2019). Os codificadores óticos sinusoidais partem desta ideia de dois sinais desfasados 90° elétricos. A grande diferença é que esses sinais não são convertidos em ondas quadradas por comparadores, mas sim convertidos sinais digitais por um ADC, aproveitando assim diferentes graus de “cinzento” em vez de apenas sinais de alto e baixo nível. Enquanto que no codificador incremental em quadratura tinha-se uma resolução angular de 360° a dividir pelo número de ciclos elétricos (ou secções transparentes/opacas) vezes quatro, nos codificadores sinusoidais ter-se-á uma resolução de 360° a dividir pelo número de ciclos elétricos a multiplicar pela resolução do conversor analógico digital (Colao, 2020).

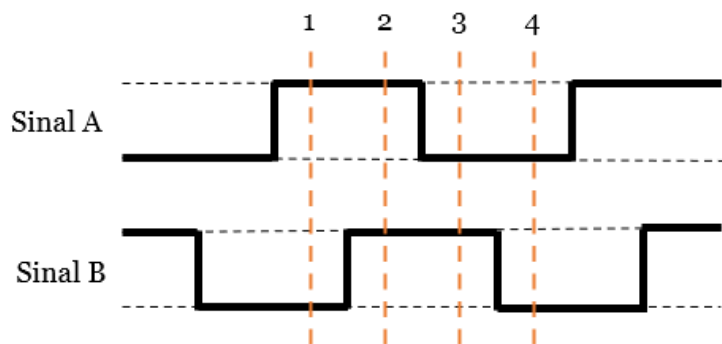


Figura 1 - Quadratura de dois sinais quadrados desfasados 90°, adaptado de(TI, 2019).

Há ainda a explicar como é calculado o ângulo neste tipo de codificadores. Uma vez que os sinais estão em quadratura, isto é, estão desfasados 90° elétricos, pode-se considerar um sinal como seno e outro como cosseno. Um dos métodos mais simples e bastante utilizado para o cálculo do ângulo é através do arco tangente dado pela Equação 1 (Burke *et al.*, 2000). A explicação para o método é dada pela matemática do círculo trigonométrico, que indica que um círculo trigonométrico no plano imaginário pode ser representado por coordenadas de uma função seno (eixo imaginário) e uma função cosseno (eixo real) e que através do arco tangente é possível medir o ângulo de determinado ponto do círculo em relação ao eixo real (Lyons, 2008). O ponto do círculo coincidente com o eixo real corresponderá ao valor zero do sinal seno e ao valor máximo do sinal cosseno, que pode ser considerado o início de cada ciclo elétrico.

$$\alpha = \arctan\left(\frac{\sin(\omega)}{\cos(\omega)}\right) \quad (1)$$

Portanto conseguindo-se gerar dois sinais desfasados 90°, com a mesma amplitude, para cada posição num determinado movimento angular é possível calcular a posição angular, relativamente ao ponto de partida. A aquisição da posição direta através do arco tangente

é um dos métodos mais simples para o efeito. No entanto, são frequentemente utilizadas expansões de Taylor da fórmula para que o cálculo seja processado de forma mais rápida (Burke *et al.*, 2000).

Na Figura 2 é possível observar algumas das limitações do cálculo do ângulo através do arco tangente. Uma vez que o arco tangente apenas mede a distância angular ao eixo real tem apenas a amplitude  $\pi$  radianos ( $180^\circ$ ) é comumente utilizado a função derivada da mesma que tem uma amplitude de  $2\pi$  radianos ( $360^\circ$ ) sendo assim capaz de medir um ciclo completo (Ver Figura 3). No entanto, esta função não resolve todos os problemas uma vez que é apenas capaz de medir de ciclo em ciclo, é apenas possível retirar a posição relativa ao início de ciclo (valor máximo da função cosseno e zero da função seno). De forma a obter-se apenas valores positivos, poderá somar-se o valor  $\pi$  a cada valor calculado pelo arco tangente sendo que aí estar-se-á a medir a distância angular ao semieixo real negativo, que em termos relativos não terá grande importância.

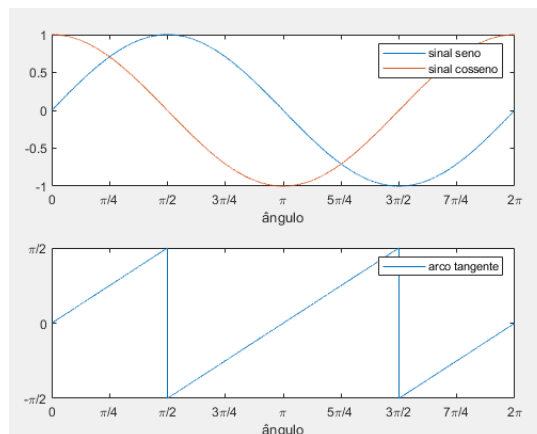


Figura 2 - Função arco tangente aplicada aos sinais seno e cosseno.

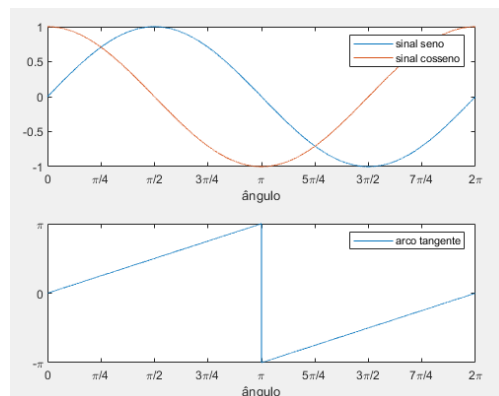


Figura 3 - Função derivada do arco tangente com amplitude de dois  $\pi$  radianos, aplicada aos sinais seno e cosseno.

### 2.1.1. Aquisição de Sinal

Em TI (2019) e Colao (2020) foram utilizados LEDs (*Light Emitting Diodes*) como fonte luminosa e foto díodos como sensor ótico, de forma a transformarem a intensidade luminosa em corrente. Também fototransistores podem ser usados como conversores de luminosidade em corrente elétrica (Lynch *et al.*, 2016). No caso do TI (2019), essa corrente é posteriormente convertida em tensão através de um amplificador de transimpedância e depois convertido em sinal digital através de um ADC.

Tanto os díodos como os transistores convencionais são constituídos por uma região P e uma região N. Sendo que os fototransistores terão ainda uma região extra P ou N, designando-se PNP ou NPN, estes terminais designam-se coletor, base e emissor. Enquanto a região N é constituída por materiais que possibilitam aos átomos ter um elétron extra na camada de valência que se poderá mover-se livremente, a região P é constituída por materiais que farão com que com a partilha de eletrões faltará um elétron para completar a camada de valência. Na fronteira destas regiões, conhecida como junção PN, parte dos eletrões em excesso da região N mover-se-ão para ocupar os espaços por ocupar pelos eletrões em falta na fronteira da região P, deixando assim espaços por ocupar na fronteira da região N, criando assim uma pequena diferença potencial nesta zona que previne que mais eletrões se movam entre as regiões. Ao conectar-se uma fonte de tensão a um díodo, parte positiva à região P, e parte negativa à região N, capaz de vencer a diferença potencial na junção PN, os eletrões conseguirão mover-se da região N para a região P, provocando a passagem de corrente no sentido inverso. O valor de tensão capaz de vencer a diferença potencial na junção PN é chamado de Tensão de rotura. Num díodo ideal, ligado a uma fonte de tensão variável, com o terminal positivo conectado à região P e o terminal negativo ligado à região N, a evolução da intensidade de corrente que passa por este elemento será aproximadamente como na Figura 4. Será zero até perto da tensão de joelho. Perto da tensão de joelho começa a haver passagem de corrente, embora os valores sejam bastante próximos de zero e em valores de tensão superiores à tensão de joelho o valor de corrente começa a ser significativo e passa a ter um comportamento quase linear. No caso de um transistor NPN, a base e emissor têm o mesmo funcionamento de um díodo sendo que quando é ultrapassada a tensão de joelho, o componente também permitirá passagem de corrente do coletor para o emissor, sendo a corrente do emissor uma soma das correntes do coletor e base. A intensidade de corrente na base terá um comportamento similar à corrente num díodo em relação à fonte de alimentação conectada à base. A corrente do

coletor é diretamente proporcional à corrente da base multiplicada pelo ganho, valor que varia de transistor para transistor (Malvino & Bates, 2016).

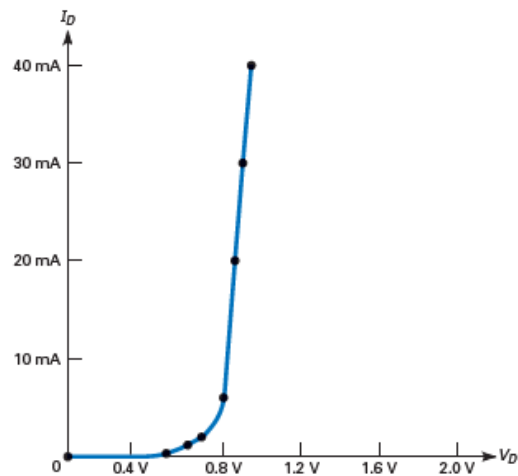


Figura 4 - Intensidade de corrente num diodo ideal com tensão de joelho de 0,8V, adaptado de (Malvino & Bates, 2016).

No caso dos fotodiodos, estes não são revestidos por uma camada opaca sobre as regiões PN e a incidência de luminosidade sobre a junção PN provocará que os elétrons transitem da região P para região N, provocando a passagem de corrente elétrica no sentido inverso ao convencional. Esta corrente não depende de uma fonte de tensão, no entanto tem valores bastante reduzidos (na ordem dos microamperes) que normalmente são contornados pela utilização de amplificadores operacionais (Lynch *et al.*, 2016). Os fototransistores utilizam um fotodiodo como conector base, permitindo que seja a intensidade luminosa a funcionar como gatilho de passagem de corrente da base e coletor para o emissor. Uma vez que a corrente de saída (corrente do emissor) depende também de uma tensão de alimentação, esta terá valores na ordem dos miliamperes, bastante superiores aos apresentados em fotodiodos (Lynch *et al.*, 2016).

Sendo por vezes mais prático ler-se valores de tensão (ADC), um dos métodos mais comuns de converter a corrente em tensão é através de divisores de tensão. Um circuito divisor de tensão é um circuito constituído por uma fonte de tensão e duas resistências como exposto na Figura 5. Sendo que a tensão de saída e a tensão entre os terminais de R2 é a mesma. Pela lei de Ohm que:

$$I = \frac{V}{R_{total}} = \frac{V}{R1+R2} \quad (2)$$

Logo:

$$V_{R2} = R2 * I = \frac{R2}{R1+R2} * V \quad (3)$$

Funcionando o transistor como uma resistência variável (embora não de forma completamente linear), ao ser-se aplicado ao circuito uma resistência entre o emissor e à referência de potencial zero, a tensão entre estes dois elementos vai variar em conformidade com a resistência do transistor, isto é, em conformidade com a corrente do emissor (Robbins & Miller, 2012; Lynch *et al.*, 2016).

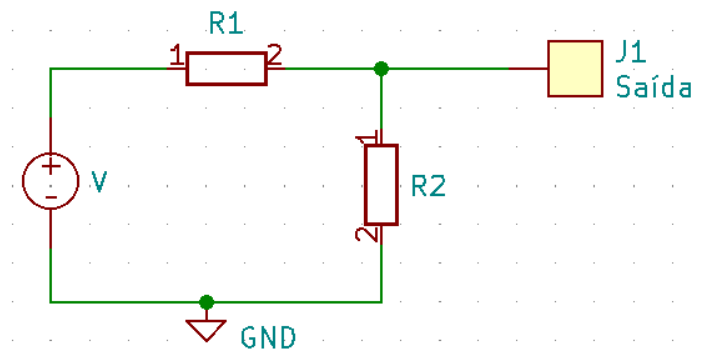


Figura 5 - Circuito divisor de tensão adaptado de (Robbins & Miller, 2012).

### 2.1.2. Conversão Analógico-Digital

Os sensores têm a responsabilidade de adquirir dados físicos do mundo real e converter esses dados em variáveis elétricas (intensidade de corrente ou diferença potencial). É o caso dos fotodiodos e fototransistores que são capazes de traduzir uma dada gama de irradiância luminosa em corrente elétrica ou tensão com o auxílio de divisores de tensão ou amplificadores de transimpedância. No entanto, dados traduzidos em corrente elétrica ou tensão para serem analisados necessitam de ser convertidos em dados digitais. Esta conversão é o que se denomina conversão analógico-digital (Kester, 2005).

Para se perceber este tipo de conversão é necessário primeiro entender algumas características como: resolução, tensão de referência e frequência de amostragem e conversão. Todos os conversores analógico-digital têm uma tensão de referência que pode ser interna ou externa. Esta tensão de referência definirá o teto máximo de tensão que o conversor será capaz de ler, assim como a referência de potencial zero definirá o

teto mínimo. A resolução (dada em bits) define a quantização do teto máximo. Ou seja, se um ADC tiver uma resolução de 4 bits, o ADC será capaz de converter a tensão recebida em 16 níveis diferentes ( $2^4$ ). Recebida a tensão de entrada, é convertida para uma palavra binária de 4 bits, neste exemplo. Uma vez que o sinal recebido pelo ADC é convertido num sinal discreto ter-se-ão sempre erros de quantização, isto é, para valores de tensão intermédios, a conversão será sempre aproximada ao nível inferior ou superior provocando alguma inexatidão. A frequência de conversão refere-se à periodicidade a que é feita a conversão. Esta característica tem bastante relevância quando o ADC está a trabalhar com um sinal variável, uma vez que se a frequência de conversão for demasiado baixa, o sinal digital não captará as principais características do sinal analógico. Pelo critério de Nyquist para não se correr este risco, a frequência de conversão deve ser no mínimo 2 vezes maior que a frequência do sinal analógico (Kester, 2005).

Antes de ser feita a conversão, o sinal analógico passa por um circuito de amostragem e espera demonstrado na Figura 6. Assim que é ativado o gatilho se iniciar a conversão, o primeiro interruptor conecta-se ao circuito até carregar o condensador. Assim que este está carregado, o primeiro interruptor desconecta-se e conecta-se o segundo interruptor iniciando a conversão pelo descarregamento do condensador. Isto faz com que a frequência de conversão tenha de ter sempre em conta o tempo que o condensador do circuito de amostragem e espera seja carregado. Embora seja muitas vezes conveniente que a frequência de conversão seja alta, deve assumir um compromisso com o carregamento do condensador (Kester, 2005).

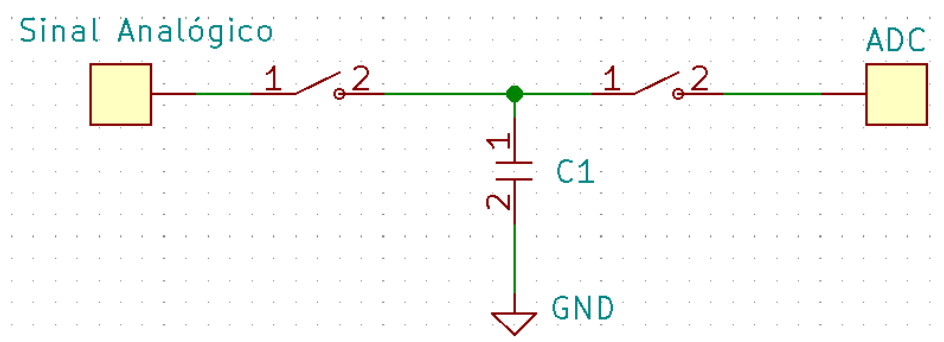


Figura 6 - Circuito de amostragem e espera adaptado de(Kester, 2005).

A conversão em si de analógico para digital pode ser feita de vários métodos dependendo do tipo de conversor. Um dos tipos de conversor mais utilizados é o conversor por aproximação sucessiva. Neste tipo de conversão, o sinal analógico estará conectado a um comparador com duas entradas. Na outra entrada do comparador estará conectado um

conversor digital para analógico que na primeira aproximação terá como saída (entrada do comparador) a tensão correspondente a uma palavra binária com todos os bits a zero exceto o bit mais significativo. No exemplo de um conversor com uma resolução de 4 bits com tensão de referência 3,3V, na primeira aproximação utilizar-se-á 1000 (8 em decimal) que corresponderá a 1,76V. Se a tensão do sinal analógico for superior a essa tensão é guardado o bit mais significativo da conversão como 1 e de seguida é comparado com a tensão equivalente à palavra 1100. Se a tensão do sinal analógico for inferior é guardado o bit mais significativo da conversão como zero e de seguida é comparado com a tensão equivalente à palavra 0100. Este processo é repetido até se chegar ao bit menos significativo. Este processo também requer algum tempo pelo que a frequência de conversão também ter de assumir um compromisso com este tempo requerido (Kester, 2005).

### **2.1.3. Ruído e Análise de Fourier**

A transformada de Fourier foi introduzida pelo matemático e físico Josef Fourier em 1807. O mesmo afirmava que qualquer sinal periódico poderia ser representado pela soma de vários sinais sinusoidais. Inicialmente rejeitado este estudo, acabou por ser aceite mais tarde. Apesar de Fourier não estar completamente correto, um sinal periódico pode não ser perfeitamente representado por um conjunto de ondas sinusoidais, pode-se obter uma representação bastante próxima da original, se suficientes ondas forem escolhidas (AD, 2003).

Este conceito veio permitir conhecer a magnitude de cada frequência do sinal, já que quando o sinal original é simplificado em sinusoides simples, cada uma delas terá uma frequência e amplitude diferentes. Isto torna-se bastante útil no que ao dimensionamento de filtros diz respeito, uma vez que permite conhecer a magnitude de frequências indesejadas (AD, 2003).

Quando um sinal contínuo é aplicado à transformada de Fourier (Equação 4), o resultado é o que se denomina de série de Fourier, função decomposta em vários sinais sinusoidais somados uns aos outros.

$$F(f) = \int_{-\infty}^{+\infty} g(t) * e^{-j*2*\pi*f*t} dt \quad (4)$$

No entanto, para sinais discretos é utilizada uma adaptação desta transformada conhecida como transformada de Fourier discreta, conforme exposto na Equação 5. Quando um conjunto de dados discretos é aplicada a esta transformada, o resultado são números complexos que não são mais que a representação das várias frequências no eixo complexo. Aplicado o módulo a cada resultado é possível saber a magnitude dessa frequência no sinal original (Ahmed, 2015).

$$F_k = \sum_{n=0}^{N-1} x_n * e^{-\frac{j*2*\pi}{N}*k*n} \quad (5)$$

A transformada de Fourier discreta é uma fórmula bastante extensiva e difícil de calcular, por isso mesmo foram criados inúmeros algoritmos para calcular a transformada computacionalmente. O algoritmo mais utilizado é conhecido como FFT (*Fast Fourier Transform*). (Ahmed, 2015)

No entanto, a transformada de Fourier discreta apresenta algumas limitações. Uma delas denominada fuga espectral. Esta limitação ocorre devido à FFT e DFT (*Discrete Fourier Transform*) assumirem que o último dado do sinal a processar seja equivalente ao dado imediatamente anterior ao primeiro dado do sinal a processar. Ao ser colocado a parte final da amostra em questão ao lado do início dessa mesma amostra, se essa amostra não corresponder a um número inteiro de ciclos, encontrar-se-á uma descontinuidade. A DFT ou FFT reproduzirá essa descontinuidade em harmônicos que não se encontram no sinal original. De forma a minimizar estes defeitos foram desenvolvidas diferentes funções que quando aplicadas ao sinal fará com que os valores iniciais e finais do mesmo tenham o valor zero, como na Figura 7. Algumas das funções mais usadas são a função Hamming, Blackman, Hanning e BlackmanHarris (Kester & Analog Devices, 2003).

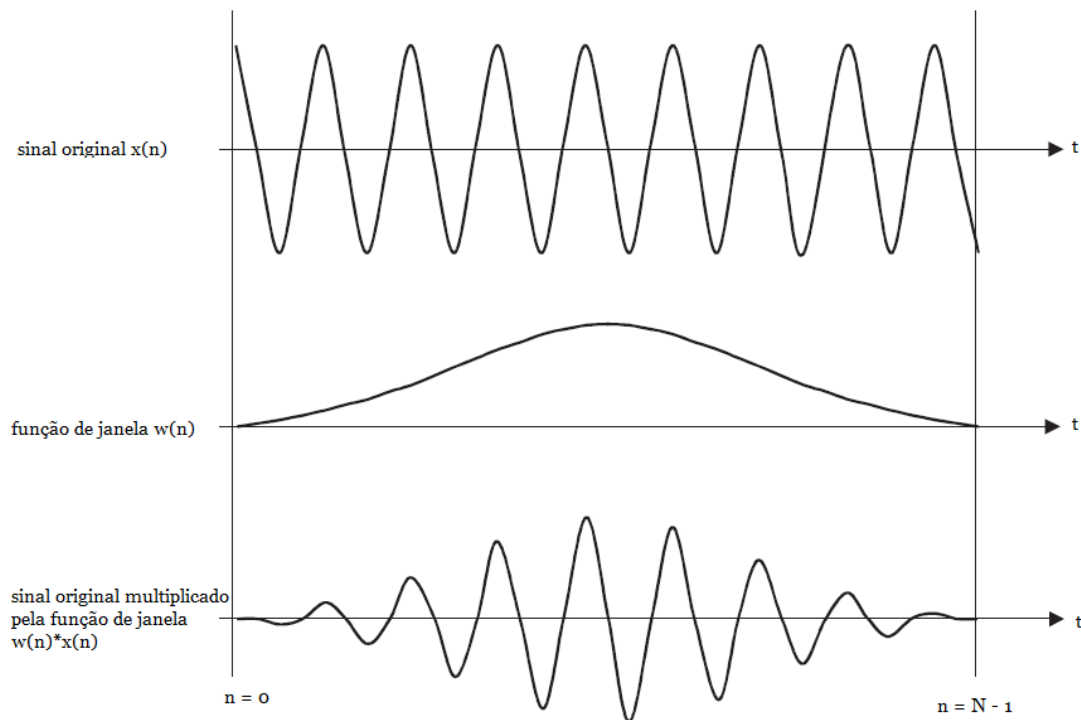


Figura 7 – Efeitos de uma função janela num sinal. Adaptado de (Kester & Analog Devices, 2003).

#### 2.1.4. Filtros Analógicos

Filtros analógicos podem definir-se como circuitos eletrónicos que têm como propósito reduzir a magnitude de determinadas gamas de frequência de um sinal analógico contínuo (Su, 2002). A classificação destes filtros pode dividir-se em filtros passivos e filtros ativos. A diferença passa apenas pelo facto dos filtros passivos não necessitarem de amplificadores ativos e energia externa para o seu funcionamento. Tanto os filtros ativos como passivos subdividem-se em várias classificações em conformidade com a gama de frequências que atenuam. Sendo que os filtros ativos podem ainda subdividirem-se de forma diferente consoante o método de dimensionamento. A subdivisão em conformidade com atenuação de frequência é feita da seguinte forma (Davis, 2017):

- Filtro passa-baixo – atenuam a amplitude de frequências elevadas
- Filtro passa-alto – atenuam a amplitude de frequências baixas

- Filtro passa-banda – permite a passagem de frequências dentro de uma determinada gama
- Filtro de rejeita-banda – atenuam a amplitude de frequências dentro de uma determinada gama (Davis, 2017)

A capacidade de reduzir amplitudes de certas gamas de frequência é conseguida devido à dependência da impedância em relação à frequência dos condensadores e indutores. Podendo-se definir a impedância de um indutor e de um condensador, respetivamente, pelas seguintes equações, onde C representa a capacitância, L a indutância e  $\omega$  a frequência angular elétrica (Zumbahlen, 2008).

$$Z_L = j * \omega * L \quad (6)$$

$$Z_C = \frac{1}{j * \omega * C} \quad (7)$$

Na Figura 8 pode observar-se o esquema elétrico de um filtro passa-baixo que passará a ser analisado.

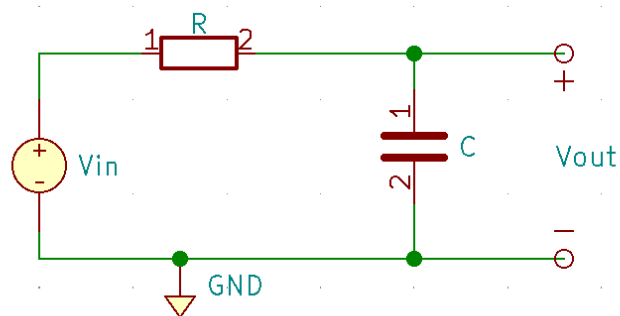


Figura 8 - Filtro RC (Adaptado de Robbins & Miller, 2013).

Aplicando o conceito de divisor de tensão ao circuito obtém-se a equação:

$$V_{out} = \frac{Z_C}{R + Z_C} * V_{in} \quad (8)$$

Através da função de transferência pode-se obter a relação entre o sinal de saída e o sinal de entrada. Utilizando as Equações 7 e 8 aplicadas à função de transferência obtém-se:

$$TF(\omega) = \frac{V_{out}}{V_{in}} = \frac{1}{1+j*\omega*R*C} \quad (9)$$

Sendo  $\omega$  a frequência angular em radianos por segundo,  $R$  o valor da resistência [ $\Omega$ ] e  $C$  a capacidade do condensador [F].

Tratando-se de uma análise no plano complexo é possível calcular o ganho entre relação entre o sinal de entrada e saída através da distância ao eixo e o desfasamento angular entre os dois sinais através da inversa da tangente. Nota: para este tipo de análises é útil o uso do diagrama de Bode que consiste num gráfico com valores logarítmicos no eixo horizontal e valores lineares no eixo vertical em dB (unidade logarítmica). Pode-se assim calcular o ganho e o desfasamento através das seguintes equações:

$$ganho = \frac{1}{\sqrt{1+(j*\omega*R*C)^2}} \quad (10)$$

$$desfasamento\ angular = -atan(\omega * R * C) \quad (11)$$

Através destas equações é facilmente observável que o ganho será menor quanto maior a frequência. Isto é, a atenuação da magnitude será maior quanto maior for a frequência pelo que se trata de um filtro passa-baixo. O mesmo acontece com o desfasamento: terá valores mais significativos para frequências altas. Também é observável que para um valor fixo de frequência angular, o ganho será quanto menor quanto maior for o valor da resistência e da capacidade do condensador. De forma inversa, o desfasamento para uma frequência angular fixa será maior quanto maior for o valor da resistência e da capacidade do condensador.

É chamada de frequência de corte a frequência em que o ganho do sinal é de -3dB em relação ao sinal de entrada e o desfasamento é de  $-45^\circ$  e pode ser calculada através da seguinte equação:

$$\omega_c = \frac{1}{R*C} \quad (12)$$

Definindo a frequência de corte como 1000Hz ( $1000*2*\pi$  rad/s), através das Equações 10 e 11 obtém-se os diagramas de Bode da resposta em frequência e angular da Figura 9 e da Figura 10.

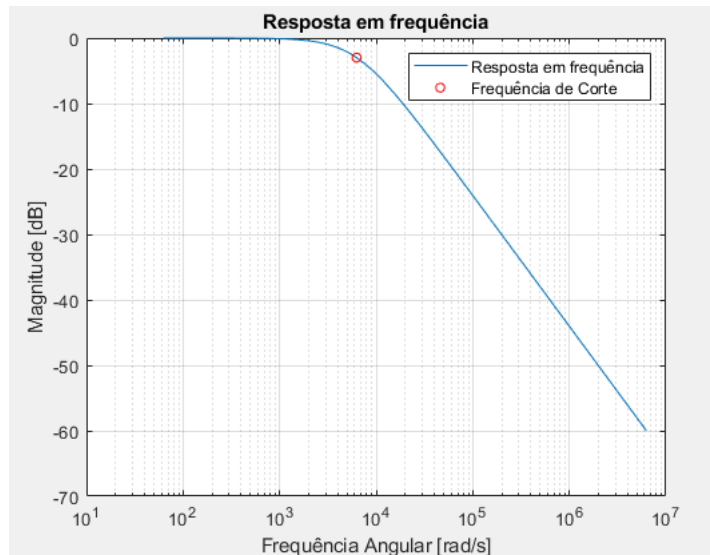


Figura 9 – Resposta em frequência (Adaptado de Robbins & Miller, 2013).

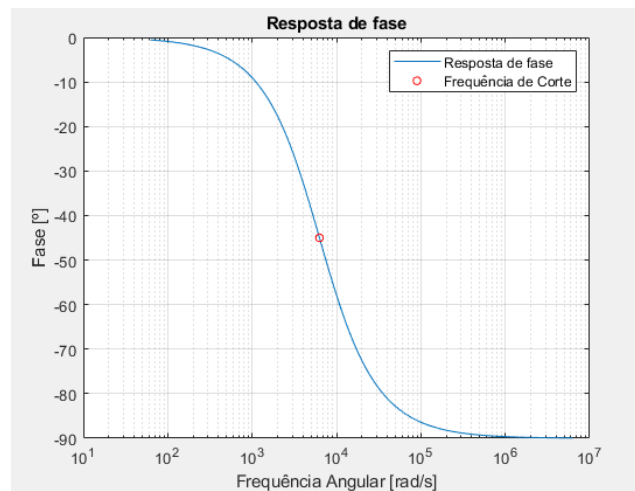


Figura 10 – Resposta de fase (Adaptado de Robbins & Miller, 2013).

### 2.1.5. Filtros Digitais

A implementação de filtros digitais tem na sua generalidade o mesmo propósito da implementação de filtros analógicos. No entanto, estes acarretam a vantagem de serem aplicados ao sinal depois da conversão analógico-digital, o que lhes poderá permitir a eliminação de erros associados à conversão ou amplificação do sinal por amplificadores operacionais (AD, 2003).

Este tipo de filtros pode definir-se como uma função discreta que quando aplicada a um sinal discreto atenua a magnitude de determinadas frequências. Esta função pode ter

uma ordem N em que cada elemento da ordem terá um valor de coeficiente definido. O valor da ordem e dos coeficientes dependerão da aplicação para que se pretende. Também estes filtros causarão desfasamento do sinal em determinadas frequências (AD, 2003).

Existem dois tipos de filtros digitais: FIR (*Finite Impulse Response*) e IIR (*Infinite Impulse Response*). Os filtros FIR apesar de apresentarem um desempenho difícil de atingir pelos filtros IIR ou filtros analógicos, requerem um elevado número de coeficientes para um bom desempenho. Já os filtros IIR, apesar de terem um desempenho semelhante aos filtros analógicos, por fazerem uso do *feedback* não tendem a necessitar de tantos coeficientes como os filtros FIR para um bom desempenho.

Nas Equações 13 e 14 estão representadas as fórmulas de cálculo do sinal filtrado através dos filtros FIR e IIR, respetivamente. As variáveis b e a correspondem aos coeficientes, x ao sinal de entrada e y ao sinal de saída (sinal filtrado) (AD, 2003).

$$y[n] = b[0] * x[n] + b[1] * x[n - 1] + b[2] * x[n - 2] \quad (13)$$

$$y[n] = b[0] * x[n] + b[1] * x[n - 1] + b[2] * x[n - 2] - a[1] * y[n - 1] - a[2] * y[n - 2] \quad (14)$$

O filtro IIR baseia-se nos princípios da transformada de Laplace e de função de transferência onde os pólos representam as raízes do denominador polinomial e os zeros representam a raiz do numerador polinomial, no plano complexo. Enquanto os zero representam as frequências onde o filtro atenua o sinal de entrada, os pólos serão responsáveis por indicar as frequências onde o filtro responderá com uma rejeição ou amplificação acentuada ao sinal de entrada (Lyons, 2011). Os filtros IIR podem dividir-se em *Butterworth*, *Chebyshev*, *Eliptic* e *Maximally flat* que consistem em métodos de dimensionamento dos filtros para diferentes aplicações (AD, 2003).

Dentro dos métodos podem-se utilizar diversas técnicas de forma a dimensionar os coeficientes destes filtros como a técnica de invariância do impulso, técnica da transformada bilinear e o método da transformada de z correspondente (AD, 2003). No entanto, nos dias que correm, *softwares* como o Matlab possuem ferramentas que permitem dimensionar estes filtros com relativa facilidade, permitindo escolher o número de coeficientes pretendido, a resposta em magnitude, a frequência de corte, etc. (Mathworks, n.d.).

## 3. Materiais e Métodos

A presente dissertação tem por objetivo o desenvolvimento de um sistema de posicionamento com codificador ótico. O sistema é baseado no projeto, desenvolvido pela *Haddington Dynamics*, “Dexter 1” (Newton, 2021).

### 3.1. Microcontrolador PIC32MZ

PIC32MZ é uma família de microcontroladores programáveis (em C ou *Assembly*) de 32 bits da *Microchip*. Esta família de microcontroladores possui um *core* com uma frequência até 252MHz, uma memória *RAM (Random Access Memory)* até 512Kb e um tamanho de memória programável até 2045Kb, consoante o modelo. Dispõe de inúmeros recursos como *Timers*, comparadores de entrada e saída, conversores analógico-digital e digital-analógico, interfaces de gráficas como *EBI (Evidence Based Intervention)* e *Direct Memory Access (DMA)*, interfaces de comunicação como *SPI (Serial Interface Communication)*, *I<sup>2</sup>C (Inter-Integrated Circuit)*, *UART (Universal Asynchronous Receiver-Transmitter)* e *CAN (Controller Area Network)*, bem como pinos configuráveis apenas como entradas ou saídas a 3,3V em nível alto. É possível aceder aos recursos através de registos, explicados mais à frente (Microchip Technology Inc., 2021).

Cada recurso tem disponíveis vários registos que possibilitam a configuração de determinado recurso. Os registos têm uma organização semelhante às estruturas em linguagem C. Isto é, cada registo ocupa um determinado espaço na memória do microcontrolador, sendo que esse espaço está subdividido em vários subregistos, podendo cada um destes ocupar um ou mais bits de memória. A configuração pode ser feita diretamente no registo, atribuindo diretamente o valor correspondente a determinada configuração para todos os subregistos ou feita individualmente em cada subregisto utilizando uma convenção semelhante às estruturas. A Figura 11 apresenta que determinado registo ocupa 10 bits na memória, subdividido em vários subregistos, começando no endereço 0 (zero) e acabando no endereço 10 (000A em hexadecimal) (Microchip Technology Inc., 2021).

Endereço de memória	Descrição	Posição (bits)
0x000A	SUBREG1	10
	SUBREG2	9
		8
		7
	SUBREG3	6
		5
		4
0x0000	SUBREG4	3
	SUBREG5	2
	SUBREG6	1
		0

Figura 11 - Exemplo de registo e correspondentes subregistos num PIC32MZ.

### 3.1.1.1. Rotina de Serviço de Interrupção (RSI)

A maioria dos recursos de que se apresentará posteriormente tem a possibilidade de ativar rotinas de serviço de interrupção. Este recurso é nada mais que uma funcionalidade (que pode ou não ser ativada) da maioria dos outros recursos. A rotina de serviço de interrupção permite que a dado momento o processamento do código seja “pausado” e seja chamada a função de interrupção. A função de interrupção tem a seguinte forma genérica: `void __attribute__ (vector("vetor do recurso"), interrupt(IPL7SRS), nomips16) "nome da interrupção"();` em que o que está entre aspas deverá ser substituído pelo que se pretende. O “vetor do recurso” terá um nome específico para cada módulo enquanto que o “nome da interrupção” será o nome que pretendemos atribuir à função de interrupção e pode ser atribuído o nome que se desejar. A função de interrupção é chamada quando é ativada a *flag*. Uma vez chamada a função de interrupção, o código nela contido, caso a *flag* não seja desativada no fim da função de interrupção, assim que o todo o código nela contido seja processado, a interrupção inicializará de novo. Norma geral, o código contido pela função deve ser curto e de fácil processamento para que o código principal não esteja muito tempo “pausado”. O momento em que é chamada a interrupção dependerá de recurso para recurso. Por exemplo, no caso dos *timers* a função de interrupção é chamada no fim de cada ciclo.

Cada recurso terá à disposição registros com diferentes funcionalidades para interrupção: habilitação, *flag* e priorização. Embora sejam apenas 3 funcionalidades diferentes, existem vários registros e subregistro para a mesma funcionalidade de forma a estarem distribuídos pelos diferentes recursos. No subcapítulo de cada recurso poderá observar-se os subregistro que lhe estão atribuídos.

O registro de habilitação,  $IECx$  ( $x$  corresponde a um número que dependerá do recurso) é o que permite habilitar a interrupção para dado recurso. O registro *flag*,  $IFs_x$  ( $x$  corresponde a um número que dependerá do recurso), é ativado automaticamente quando o evento de gatilho da interrupção acontece, ao ser ativado permite a chamada da função de interrupção. Como dito, este registro deve ser desativado pelo *software* no fim da função de interrupção para que o processamento normal do código possa ser retomado. Os registros de priorização,  $IPC_x$  ( $x$  corresponde a um número que dependerá do recurso), estão divididos em dois subregistro diferentes: um subregistro de priorização e outro de subpriorização. A estes subregistros deve ser atribuído um valor de 1 a 7 que corresponde à prioridade de determinada interrupção (1 corresponde ao valor de maior prioridade e 7 ao valor de menor prioridade). Estes subregistro têm como objetivo quando duas interrupções são chamadas no mesmo momento definir qual a interrupção que deve ser executada primeiro. Quando duas interrupções têm o mesmo nível de prioridade recorre-se à subprioridade para se definir qual a interrupção a executar primeiro (Microchip Technology Inc., 2015a).

### **3.1.1.2. Oscilador**

A família de microcontroladores PIC32MZ possui um relógio principal denominado de  $SYSClk$  que servirá como fonte de relógio a inúmeras funcionalidades (isto é, determinará a periodicidade de acontecimentos de diversas funcionalidades). A frequência de funcionamento deste relógio deriva de um dos 5 osciladores existentes no microcontrolador. Estes 5 osciladores estão divididos em 3 osciladores internos (FRC, BFRC e LPRC) e 2 osciladores externos (POSC e SOSC) (Microchip Technology Inc., 2021). Os osciladores externos são compostos por um circuito elétrico com um oscilador de cristal que está conectado ao microcontrolador, no caso do POSC através dos pinos OSC1 e OSC2, e no caso do SOSC através dos pinos OSCI e OSCO. Tanto os osciladores internos como os osciladores externos permitem vários modos de funcionamento configuráveis através de inúmeros registros. Através dos registros é também possível

configurar o oscilador que servirá como fonte de relógio para o relógio principal SYSCLK, por predefinição, se os valores dos registos não forem alterados, SYSCLK derivará de FRC. Na secção 42 do manual de referência dos PIC32 é dado um exemplo (exemplo 42-2 do manual de referência) de configuração de forma a atingir uma frequência de relógio de 200MHz (Microchip Technology Inc., 2013b).

No entanto, o relógio SYSCLK não serve diretamente como fonte de relógio a todas as funcionalidades do microcontrolador. Muitas funcionalidades utilizam como fonte de relógio PBCLKx que são 8 fontes de relógio que derivam do SYSCLK (x representa um número de 1 a 8). Na ficha técnica do PIC32MZ é apresentada uma tabela (tabela 8-1 da ficha técnica) que demonstra que fontes de relógio cada funcionalidade poderá usar. Nos registos correspondentes a cada funcionalidade será possível escolher a fonte de relógio pretendida. Para a utilização destas fontes será ainda necessário configurar-se os registos PBxDIV onde é possível habilitar a fonte de relógio e ainda dividir a frequência do SYSCLK para a nova fonte de relógio de 1 até 128 (Tabela A1) (PIC, n.d.-a).

### **3.1.1.3. Portas de Entrada e Saída**

As portas de entrada e saída permitem ao microcontrolador comunicar com o mundo exterior recebendo ou enviando informação. As portas estão divididas em pinos, cada pino representa uma entrada/saída, ou seja, cada porta representa um conjunto de entradas/saídas. As portas são designadas por letras e os pinos designados por números, antecidos de RP. Por exemplo RPE representa uma porta e RPE7 representa um pino dessa mesma porta.

A configuração dos pinos como entradas e saídas funciona através dos registos TRIS<sub>x</sub> e respetivo subregisto do pino que se quer configurar. Ao atribuir-se o valor 1 ao subregisto, o pino é configurado como entrada, ao atribuir-se o valor 0 ao subregisto, o pino é configurado como saída. O formato de escrita do subregisto para a configuração de determinado pino funcionará da seguinte forma: TRIS<sub>xbits</sub>.TRIS<sub>xy</sub>, em que x representa a letra da porta que se pretende configurar e y o número do pino que se pretende configurar. Ou seja, para se configurar o pino RPA5 como entrada deveria escrever-se TRISA<sub>bits</sub>.TRISA5 = 1.

Para os pinos configurados como saída é possível atribuir-se-lhe um nível lógico que corresponderá em caso de nível alto a 3,3V. Para isso é necessário utilizar os registos `LATx` que funcionam da mesma forma que os registos apresentados anteriormente. Para se atribuir nível alto ao pino `RPG3` (tem de ter sido anteriormente configurado como saída) deve escrever-se no compilador `LATGbits.LATG3 = 1`.

É ainda possível configurar os pinos como analógicos ou digitais (requerido para algumas funcionalidades). Por defeito, todos os pinos estão configurados como pinos analógicos. O registo que permite configurar é o registo `ANSELx` que funciona da mesma forma que os anteriores. A atribuição de 0 ao subregisto configura o respetivo pino para pino digital e a atribuição de 1 configura o pino para pino analógico. (Microchip Technology Inc., 2015b)

Alguns dos pinos do microcontrolador estão multiplexados de forma a permitir que não só funcionem como entradas ou saídas mas também sirvam de entradas ou saídas para outras funcionalidades como comparadores de entrada, comparadores de saída, UART, ADC, etc.. O mapeamento dos pinos que funcionam como entrada e que funcionam como saída para determinadas funcionalidades está representado nas tabelas 12-2 (para entradas) e 12-3 (para saídas) da ficha técnica do PIC32MZ (Microchip Technology Inc., 2021). Para o caso da atribuição de um pino como entrada de uma determinada funcionalidade é necessário escrever no compilador a designação da funcionalidade descrita na coluna “[pin name]R\_SFR” e igualar ao valor do pino pretendido, demonstrado na última coluna (nota: o valor está em binário, para uma escrita em binário no compilador, o valor deve ser antecedido de “0b”). Para o caso da atribuição de um pino como saída a situação é a inversa, é necessário escrever no compilador a designação do pino pretendido e igualar ao valor da funcionalidade pretendida, demonstrado na última coluna. (Microchip Technology Inc., 2021)

#### **3.1.1.4. Temporizador**

O PIC32MZ tem à sua disposição 9 temporizadores (*Timers*) de 16 bits, sendo que o *Timer 1* tem ligeiras diferenças dos restantes. A principal diferença prende-se pelo facto dos *Timers 2* ao *9* permitirem combinar-se com o *Timer* seguinte (isto é, o *Timer2* pode combinar-se com o *Timer3*, o *Timer4* com o *Timer5*, etc.) e formar um *Timer* de 32 bits (Microchip Technology Inc., 2021). O *Timer 1* tem ainda a desvantagem de ter menos um

bit no subregistro `TCKPS` (Tabela A3), impossibilitando os mesmos níveis de subdivisão do contador dos restantes *Timers*. Para a configuração dos *Timers* são utilizados os registros `TxCON` e `PRx`, existindo ainda o registro `TMRx` que funciona como contador (Nota: `x` corresponde ao número do timer a configurar).

Para perceber o funcionamento dos *Timers* é necessário primeiro perceber como funcionam os registros `TMRx` e `PRx` (`x` deve ser substituído pelo número do Timer a utilizar). O contador `TMRx` é incrementado à frequência do `PBCLKx` a dividir pelo valor de pré-escala (subregistro `TCKPS`). Ou seja, se a frequência `PBCLKx` for 100MHz e for atribuído o valor 1 ao subregistro `TCKPS`, para os *Timers* 2 a 9, o `TMRx` vai ser incrementado a cada 50MHz. O registro `PRx` tem como propósito definir o valor máximo para o contador. Quando o contador atinge o valor de atribuído ao `PRx`, o contador é zerado e reinicia-se o processo. A incrementação desde o início até atingir o valor de `PRx` corresponde a um ciclo. Este é o método de funcionamento mais simples (Timer de 16 bits) em que os subregistros `TCS`, `TGATE` e o `T32`, para os *Timers* 2 a 9, lhes é atribuído o valor zero (`TxCONbits.TCS = 0`; `TxCONbits.TGATE = 0`; `TxCONbits.T32=0`). A frequência desejada para cada ciclo pode ser calculada através da seguinte equação:

$$Frequência = \frac{PBCLK}{PRx \cdot Préescalar} \quad (15)$$

Como dito anteriormente, é possível combinarem-se dois *Timers* para perfazer um Timer de 32bits. Para isso é necessário ativar o subregistro `T32`. Neste modo de funcionamento são combinados os registros `TMR` e `PR` de dois *Timers* formando "dois registros" de 32 bits. O `TMRx` e `PRx` correspondem aos 16 bits menos significativos e o `TMRy` e `PRy` correspondem aos 16 bits mais significativos (nota: `y=x+1`). Para efeitos de configuração os registros `PRx` e `PRy` podem ser carregados individualmente, dividindo o valor de 32 bits em dois valores de 16 bits e respeitando a significância ou pode-se carregar o `PRx` com o valor total que o compilador encarregar-se-á de atribuir os 16 bits mais significativos ao `PRy`. Posto isto, o funcionamento é muito similar ao funcionamento a 16bits, o `TMRxy` vai sendo incrementado à frequência de  $\frac{PBCLK}{Préescalar}$  e quando atinge o valor `PRxy` é zerado e reinicia-se o novo ciclo. Este método de funcionamento tem a vantagem de se poder dividir a frequência de `PBCLK` em valores menores, isto é, em períodos de ciclo maiores (Microchip Technology Inc., 2013a).

Cada *Timer* tem uma rotina de serviço de interrupção atribuída. A interrupção irá ser chamada no fim de cada ciclo do *Timer*, configurado para o efeito. Para a utilização da

rotina de serviço de interrupção deve ser atribuído ao registo de habilitação o valor de 1 e atribuído aos registos de prioridade os valores pretendidos para a priorização em relação a outras interrupções. É também de boa prática, na configuração do *Timer*, atribuir o valor zero ao registo de *flag* de forma a garantir que está a zero. Os vetores, registos e subregisto de cada timer estão representados na Tabela A5 e na Tabela A6 (Microchip Technology Inc., 2021).

### 3.1.1.5. Comparador de saída

De modo generalista, o comparador de saída tem como propósito fazer variar uma saída entre nível alto e nível baixo periodicamente. Para isto, cada módulo de comparador de saída tem várias saídas possíveis de serem associadas, Tabela 12-3 (para saídas) da ficha técnica do PIC32MZ (Microchip Technology Inc., 2021). A periodicidade da variação é feita com o auxílio dos módulos Timer sendo que cada módulo de OC (*Output Compare*) tem *Timers* associados como demonstrado na Tabela A7. Este módulo possui 9 comparadores de saída e 3 diferentes registos para a sua configuração:  $OC_xCON$ ,  $OC_xR$  e  $OC_xRS$  ( $x$  corresponde ao número do OC a configurar). O  $OC_xCON$  serve para efetuar a configuração de saída em conformidade com a Tabela A8. Os registos  $OC_xR$  e  $OC_xRS$  servem para fazer uma comparação com o  $TMR_x$  e alterar o estado conforme o modo de funcionamento selecionado (dependendo do modo selecionado, nem sempre são utilizados ambos os registos).

O comparador de saída apresenta 3 possíveis modos de funcionamento: modo de comparação única, modo de comparação dupla e modo PWM (*Pulse Width Modulation*). Passar-se-á apenas à explicação do funcionamento do modo de comparação dupla utilizado na presente dissertação.

O modo de comparação dupla utiliza os registos  $OC_xR$  e o  $OC_xRS$  para fazer a comparação. Neste modo, o pino selecionado como saída do comparador passa a nível alto quando o registo  $TMR_x$  iguala o valor de  $OC_xR$  e volta a nível baixo quando o  $TMR_x$  iguala o valor  $OC_xRS$ . Este modo está dividido em dois submodos que apenas diferenciam na continuidade do pulso. No submodo “pulso único” o pino de saída é apenas ativado e desativado uma vez, isto é, no primeiro ciclo do *Timer*. No submodo “pulso contínuo” o pino de saída é ativado/desativado sempre que o  $TMR_x$  iguala  $OC_xR/OC_xRS$ , até que a configuração seja alterada (Figura 12).



Figura 12. Funcionamento do pino de saída em Modo de Comparação Dupla com pulso contínuo e respetiva configuração. Adaptado de (Microchip Technology Inc., 2011).

Tal como nos *Timers*, é permitido também um modo de 32 bits em que são combinados dois registos  $OCxR$  e  $OCxRS$  ( $OCxR/OCyR$  e  $OCxRS/OCyRS$ ) de dois comparadores de saída formando registos de 32 bits. O  $OCxR$  e  $OCxRS$  correspondem ao 16bits mais significativos e o  $OCyR$  e  $OCyRS$ . (Nota: para o funcionamento do comparador de saída com 32 bits também o *Timer* que define o período deverá estar combinado com um 2º *Timer*).

Cada comparador de saída tem uma rotina de serviço de interrupção atribuída. O momento em que a *flag*, e conseqüentemente a interrupção, é ativada dependerá do modo para que o comparador de saída for configurado. No modo de comparação dupla a interrupção é ativada no momento em que o pino de saída retorna a nível baixo. Para a utilização da rotina de serviço de interrupção deve ser atribuído ao registo de habilitação o valor de 1 e atribuído aos registos de prioridade os valores pretendidos para a priorização em relação a outras interrupções. É também de boa prática, na configuração do comparador de saída, atribuir o valor zero ao registo de *flag* de forma a garantir que está a zero. Os vetores, registos e subregisto de cada comparador de saída estão representados na Tabela A9 e na Tabela A10 (Microchip Technology Inc., 2011).

### 3.1.1.6. Recetor-Transmissor Assíncrono Universal

Os registos do Recetor-Transmissor Assíncrono Universal (UART – *Universal Asynchronous Receiver/Transmitter*) permitem controlar a taxa de transmissão, paridade, número de bits de paragem bem como ainda dão a possibilidade de ler o estado da transmissão através de vários subregistos. Os registos disponíveis para a configuração são UxMODE, UxSTA, UxTXREG, UxRXREG e IxBRG. Os registos UxTREG e UxRXREG têm um subregisto, cada um, de 9 bits (UxTXREGbits.TX e UxRXREGbits.RX) que servem para carregar a mensagem de envio e receção, respetivamente. Isto quer dizer que tanto a mensagem de envio como de receção não poderá ser maior que 9 bits.

É possível definir a taxa de transmissão através do subregisto UxMODEbits.BRGH e do registo UxBRG. O subregisto UxMODEbits.BRGH atribuirá um fator de divisão ao relógio destinado à UART de 4 (se 1) ou 16 (se zero). Já o registo UxBRG contém apenas um subregisto de 16 bits que servirá também como fator de divisão com o valor que lhe for atribuído. A taxa de transmissão pode assim ser calculada da seguinte forma:

$$\text{Se UxMODEbits.BRGH=0 : Taxa de transmissão} = \frac{PBCLK}{(BRG+1) * 16} \quad (16)$$

$$\text{Se UxMODEbits.BRGH=1 : Taxa de transmissão} = \frac{PBCLK}{(BRG+1) * 4} \quad (17)$$

Para o funcionamento no modo mais simples de comunicação são apenas necessários o subregisto UxMODEbits.ON (que serve para ativar a comunicação e por isso deve ser colocado a 1), o subregisto UxMODEbits.BRGH, o registo UxBRG (estes últimos dois explicados anteriormente), PDSEL e STESEL. O subregisto UxMODEbits.PDSEL tem como propósito definir o comprimento e a paridade da mensagem a enviar ou receber (Tabela A11). Já UxMODEbits.STESEL serve para definir o número de bits de paragem da mensagem (Tabela A11). Já no registo UxSTA (Tabela A12 e Tabela A13) têm-se os subregistos URXEN e UTXEN que habilitam a receção e o envio (respetivamente) de dados através do pino selecionado para a UART. Existem ainda registos apenas de leitura que nos permitem saber o estado da transmissão/receção e respetivos *buffers*. Estes subregistos são bastante úteis para o controlo pois permitem saber se a transmissão foi concluída e passar assim para modo receção ou vice-versa. Os subregistos UTXISEL e URXISEL são importantes na mesma medida uma vez que permitem ativar as interrupções em diferentes momentos da transmissão/receção.

Por último existem ainda os registos  $U_{xTXREG}$  e  $U_{xRXREG}$  que contêm os subregistos TX e RX de (9 bits cada um) que funcionam como *buffers* da transmissão e receção, respetivamente. Isto é, os dados carregados no *buffer* TX serão enviados em comunicação UART através do pino selecionado para o efeito. Assim como, através do pino selecionado como pino de receção UART o microcontrolador receberá a mensagem que carregará no *buffer* RX. Nota: se o comprimento da mensagem a enviar for definido como 8bits no subregisto  $U_{xMODEbits.PDSEL}$  apenas se poderá carregar estes *buffers* com 8 bits (Microchip Technology Inc., 2012).

A UART tem três vetores de interrupção diferentes para cada módulo: um vetor para notificar a receção, um vetor para notificar o envio e um vetor de falha. A ativação da interrupção para os primeiros dois é definida pelos subregistos  $U_{xSTAbits.URXISEL}$  e  $U_{xSTAbits.UTXISEL}$ , respetivamente. O vetor de falha servirá para ativar a interrupção caso alguma falha aconteça. Os vetores, registos e subregisto de cada comparador de saída estão representados na Tabela A14 e Tabela A15 (Microchip Technology Inc., 2021).

### **3.1.1.7. Conversão Analógico-Digital**

O PIC32MZ possui 6 módulos de conversão analógico-digital com um circuito *Sample & Hold* cada um com até 12 bits de resolução (configurável através de registos). Estes 12bits significam que o ADC consegue dividir a tensão de entrada em 4096 “passos” ( $2^{12}$ ). Isto é, se a tensão de referência (que define o teto máximo da tensão de entrada no ADC) for 3,3V o ADC será capaz de recolher intervalos de valores de tensão de 0,8058608mV ( $3,3V/4095$ ). Logicamente 0V corresponderá ao valor 0 e 3,3V corresponderá ao valor 4095. O microcontrolador possui uma referência de tensão interna de 3,3V mas pode ser configurada para uma referência de tensão externa através dos pinos RA9 e RA10 que correspondem à referência externa negativa e à referência externa positiva. Esta configuração é também feita através dos registos.

Cada um dos módulos possui duas ou mais possíveis entradas positivas, bem como a possibilidade de fazer-se a conversão em modo diferencial ou em modo simples. A configuração neste microcontrolador permite-nos ainda definir o tempo de amostragem (*sampling*), escolher qual o gatilho que provoca a conversão, selecionar as tensões de referência, entre muitas outras características. Devido à enorme complexidade deste

conversor, por questões de síntese serão apenas apresentadas e explicadas as características para os modos de funcionamento mais simples e as que foram utilizadas na presente dissertação.

Os 6 módulos de ADC (*Analog-Digital Converter*) presentes nesta família PIC32MZ são designados como ADC0, ADC1, ADC2, ADC3, ADC4 e ADC7. As entradas analógicas de cada um destes módulos estão representadas na Tabela A20. Enquanto que os módulos ADC0 ao ADC4 têm entradas dedicadas, isto é, é necessário configurar a entrada com a qual se quer trabalhar, o módulo ADC7 tem entradas partilhadas, ou seja, todas as entradas estão ligadas por predefinição e pode-se utilizar mais que uma em simultâneo (desde que as leituras respeitem os timings de amostragem e conversão). Por este motivo alguns dos registos utilizados para configuração destes módulos são diferentes. A configuração da entrada analógica nos primeiros módulos é feita através do registo ADCTRGMODE (Tabela A22).

Antes de se avançar para apresentação dos registos é necessário perceber-se alguns conceitos. Cada módulo é constituído por um circuito de amostragem e retenção (*Sampling & Hold*). Este tipo de circuito tem entre a entrada e a saída (neste caso a saída será a entrada do módulo), um condensador e dois *switches* (um de cada lado do condensador). Numa primeira fase (a fase de amostragem) o primeiro *switch* está ligado ao circuito e o segundo desligado da saída, permitindo assim carregar o condensador. De seguida, o primeiro *switch* é desligado e o segundo é conectado à saída permitindo ligar o condensador à saída (entrada do módulo), fazendo a fase de retenção em que é descarregado o condensador. A conversão de analógico para digital acontece neste último momento. Como foi mencionado anteriormente, os registos permitem-nos controlar o tempo de amostragem e retenção, bem como o gatilho que aciona a conversão. Uma vez que pode ser utilizado um relógio como gatilho é preciso diferenciar o relógio/período de gatilho e o relógio/período de conversão. Quando utilizado um relógio como gatilho as conversões são feitas de forma periódica. No início de cada período do relógio é iniciada a amostragem, terminada a amostragem é iniciada a conversão, esta última terminada o circuito fica à espera de novo início de período do relógio para iniciar nova amostragem e consequente conversão. Após terminada uma conversão pode de seguida ser iniciada uma nova amostragem caso haja mais que um canal em que se pretende fazer conversões (apenas para módulo ADC7). O relógio de conversão pode ter uma fonte de relógio ( $T_{clk}$ ) diferente do relógio de gatilho. Esta fonte de relógio passa primeiro por registos que servem como divisores de frequência (ou se

preferirmos multiplicadores de período) até “se tornar” no relógio de ADC. Este relógio ADC ( $T_{AD}$ ) vai determinar o tempo de amostragem e conversão através das seguintes equações:

- Para o ADC0, ADC1, ADC2, ADC3 e ADC4

$$t_{SAMP} = (ADCxTIMEbits.SAMC + 2) \times T_{AD} \quad (18)$$

$$t_{CONV} = ((n^{\circ} \text{ de bits de resolução}) + 1) \times T_{AD} \quad (19)$$

- Para o ADC7

$$t_{SAMP} = (ADCCON2bits.SAMC + 2) \times T_{AD} \quad (20)$$

$$t_{CONV} = ((n^{\circ} \text{ de bits de resolução}) + 1) \times T_{AD} \quad (21)$$

Ver exemplificação dos períodos de gatilho, conversão e retenção/conversão na Figura 13.

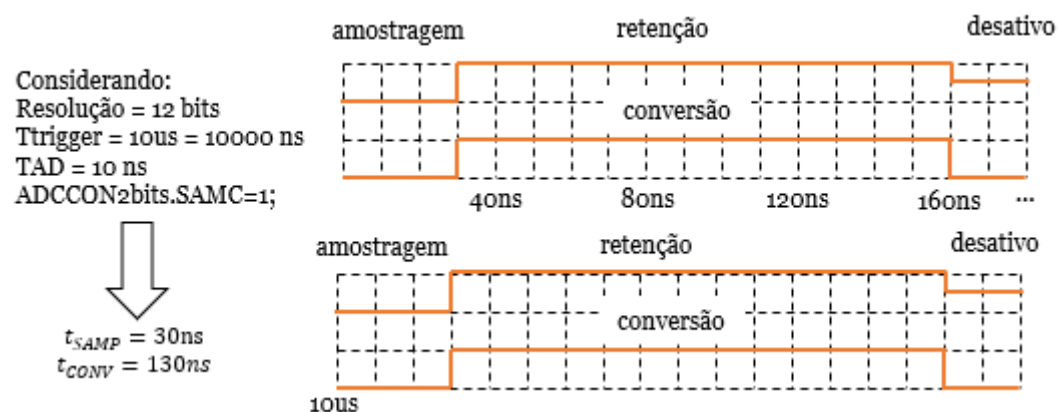


Figura 13. Representação dos períodos de gatilho, conversão e retenção/conversão.

De seguida serão apresentados os registos e respetivos registos que podem ser utilizados para um funcionamento simples e/ou utilizados na presente dissertação (Microchip Technology Inc., 2019).

### 3.1.1.7.1. ADCCON1

- $ADCCON1bits.SELRES$  – permite escolher a resolução da conversão para o módulo ADC7, entre 12, 10, 8 ou 6 bits consoante se o valor atribuído ao subregisto é 3, 2, 1 ou zero (respetivamente).

- `ADCON1bits.FRACT` – permite escolher se o valor da conversão é feito em valor inteiro ou fracionário. Valor 1: Fracional; Valor 0: Inteiro.
- `ADCON1bits.ON` – habilita o conversor quando atribuído o valor 1.

### **3.1.1.7.2. ADCCON2**

- `ADCCON2bits.BGVRRDY` – subregistro de leitura em que lhe é atribuído o valor 1 quando a tensão de intervalo de banda e a tensão de referência ADC estão prontas e o valor zero no caso inverso.
- `ADCCON2bits.REFFLT` – subregistro de leitura em que lhe é atribuído o valor 1 se a tensão de referência apresentar algum tipo de erro após o subregistro `ADC1CONbits.ON` ser habilitado. Se lhe for atribuído o valor zero após o subregistro `ADCCON1bits.ON` ser habilitado a tensão de referência não apresenta qualquer tipo de erro.
- `ADCCON2bits.CVDCPL` – permite escolher o valor do condensador do módulo ADC7 entre opF e 17,5pF, conforme a Tabela A17.
- `ADCCON2bits.SAMC` – permite escolher o tempo de amostragem ( $t_{SAMP}$ ) para o módulo ADC7 em função de TAD7, conforme a Tabela A17.
- `ADCCON2bits.ADCDIV` – permite dividir a frequência da fonte de relógio selecionada, para o módulo ADC7, em múltiplos de 2 até 254, passando a designar-se esta nova periodicidade de  $T_{AD7}$ .

### **3.1.1.7.3. ADCCON3**

- `ADCCON3bits.ADCSEL` – permite escolher a fonte de relógio ADC ( $T_{CLK}$ )
- `ADCCON3bits.CONCLKDIV` – permite dividir a frequência da fonte de relógio em múltiplos de 2 até 64, passando a nova periodicidade de relógio a designar-se  $T_Q$ .

- `ADCCON3bits.DIGENx` – permite ativar o módulo ADC que se pretende utilizar. `x` representa o número do módulo que se pretende ativar.
- `ADCCON3bits.VREFSEL` – permite escolher a tensão de referência entre  $AV_{DD}$  e  $AV_{SS}$  que correspondem às referências internas do microcontrolador positiva (3,3V) e negativa (0V) e referências externas que podem ser conectadas a um pin destinado para o efeito.
- `ADCCON3bits.GLSWTRG` – tem como função ser o gatilho da conversão. Sempre que estiver nível alto é ativado o gatilho de conversão. Nota: para a utilização deste registo é necessário seleciona-lo no subregisto `ADCTRG1bits.TRGRCx`.
- `ADCCON3bits.GSWTRG` - tem como função ser o gatilho da conversão. Sempre que passa de nível baixo para nível alto é ativado o gatilho de conversão. Nota: para a utilização deste registo é necessário seleciona-lo no subregisto `ADCTRG1bits.TRGRCx`.

#### **3.1.1.7.4. ADCxTIME**

- `ADCxTIMEbits.SELRES` - permite escolher a resolução da conversão para os módulos `ADCo`, `ADC1`, `ADC2`, `ADC3` e `ADC4`, entre 12, 10, 8 ou 6 bits consoante se o valor atribuído ao subregisto é 3, 2, 1 ou zero (respetivamente).
- `ADCxTIMEbits.ADCDIV` – permite dividir a frequência da fonte de relógio selecionada, para o módulo `ADCo`, `ADC1`, `ADC2`, `ADC3` e `ADC4`, em múltiplos de 2 até 254, conforme a Tabela A19, passando a designar-se esta nova periodicidade de `TADx`.
- `ADCxTIMEbits.SAMC` - permite escolher o tempo de amostragem ( $t_{SAMP}$ ) para os módulos `ADCo`, `ADC1`, `ADC2`, `ADC3` e `ADC4`, em função de `TADx`, conforme a Tabela A19. Nota: `x` em `ADCxTIME` corresponde ao número do módulo ADC que se pretende configurar.

### 3.1.1.7.5. ADCIMCON<sub>y</sub>

- ADCIMCON<sub>y</sub>bits.DIFF<sub>x</sub> – permite escolher entre conversão diferencial e conversão simples no canal AN<sub>x</sub>:
  - 1 – Modo conversão diferencial
  - 0 – Modo conversão simples
- ADCIMCON<sub>y</sub>bits.SIGN<sub>x</sub> – permite escolher se a conversão é feita para valores sem sinal ou com sinal no canal AN<sub>x</sub>:
  - 1 – Conversão para valores com sinal
  - 0 – Conversão para valores sem sinal

Nota: x corresponde ao canal AN<sub>x</sub> que se pretende utilizar. y pode ter valores entre 1 e 3 correspondendo aos 3 registos ADCIMCON<sub>y</sub>. O registo ADCIMCON1 contém os subregistos do canal AN<sub>0</sub> ao canal AN<sub>15</sub>, o registo ADCIMCON2 contém os subregistos do canal AN<sub>16</sub> ao canal AN<sub>31</sub> e o registo ADCIMCON3 contém os subregistos do canal AN<sub>32</sub> ao canal AN<sub>44</sub>.

### 3.1.1.7.6. ADCTRG<sub>y</sub>

- ADCTRG<sub>y</sub>bits.TRGSRC<sub>x</sub> – permite escolher o gatilho para a conversão de acordo com a Tabela A21.

Nota: x corresponde ao canal AN<sub>x</sub> que se pretende utilizar. y pode ter valores entre 1 e 3 correspondendo aos 3 registos ADCTRG<sub>y</sub>. O registo ADCTRG1 contém os subregistos do canal AN<sub>0</sub> ao canal AN<sub>3</sub>, o registo ADCTRG2 contém os subregistos do canal AN<sub>4</sub> ao canal AN<sub>7</sub> e o registo ADCTRG3 contém os subregistos do canal 8 ao canal AN<sub>11</sub>.

### **3.1.1.7.7.   ADCGIRQEN<sub>y</sub>**

- `ADCGIRQENybits.AGIENx` – permite habilitar a interrupção de determinado canal, assim que é feita a conversão e os dados são gravados

Nota:  $x$  corresponde ao canal  $AN_x$  para o qual se pretende habilitar a interrupção.  $y$  pode ter os valores entre 1 ou 2 correspondendo aos 2 registos `ADCGIRQENy`. O registo `ADCGIRQEN1` contém os subregistos do canal  $AN_0$  ao canal  $AN_{31}$  e o registo `ADCGIRQEN2` contém os subregistos do canal  $AN_{32}$  ao canal  $AN_4$  (Microchip Technology Inc., 2021).

### **3.1.1.7.8.   ADCANCON**

- `ADCANCONbits.WKUPCLKCNT` – permite definir o tempo que o módulo ADC deve demorar a aquecer para inicializar as conversões. Este tempo é definido em múltiplos de ciclo do relógio ADC conforme a Tabela A23. Este registo é comum a todos os módulos
- `ADCANCONbits.WKRDYx` – subregisto de leitura em que lhe é atribuído automaticamente o valor 1 ao subregisto assim que o módulo está pronto para inicializar. Isto é assim que é cumprido o tempo indicado no subregisto `WKUPCLKCNT`.  $x$  representa o número do módulo que se pretende verificar.
- `ADCANCONbits.ANENx` – quando ativado é iniciado o tempo de aquecimento definido pelo subregisto `ADCANCONbits.WKUPCLKCNT`.

### **3.1.1.7.9.   ADCDATA<sub>x</sub>**

`ADCDATAx` corresponde a vários registos ( $x$  corresponde mais uma vez ao número do canal  $AN_x$ ) de 32 bits que contém um subregisto `DATA` do mesmo tamanho onde são guardados os dados convertidos. Ou seja, assim que é feita a conversão os dados são armazenados em `ADCDATAxbits.DATA` até serem escritos novos dados após uma nova conversão (Microchip Technology Inc., 2021).

### **3.1.1.7.10. ADCxCFG**

É sugerido na ficha técnica do microcontrolador que os registos ADC1CFG, ADC2CFG, ADC3CFG, ADC4CFG e ADC7CFG sejam igualados a DEVADC0, DEVADC1, DEVADC2, DEVADC3, DEVADC4 e DEVADC7, respetivamente, para efeitos de calibração. (Microchip Technology Inc., 2021)

### **3.1.1.8. Rotina de Serviço de Interrupção**

Apesar do microcontrolador PIC32 apresentar inúmeros vetores disponíveis para o serviço de interrupção apenas o vetor global é relevante para a presente dissertação. A interrupção deste vetor é chamada sempre que uma conversão é concluída, permitindo carregar uma variável com o valor do *buffer* correspondente à conversão realizada. Este vetor e os respetivos registos estão representados na Tabela A20.

### **3.1.2. Placa de circuito impresso Zonicstech**

A placa de circuito impresso fornecida pela Zonicstech apresentada na Figura 14 é constituída por um PIC32MZ2048EFM100 (100 pinos). O microcontrolador está conectado a um regulador de tensão (LD1117S33CTR) que por sua vez está ligado a um terminal permitindo que a alimentação seja feita a 5V. Para a programação do microcontrolador a placa tem conectores apropriados a uma ligação com um *debugger*.

O *layout* da placa foi concebido de forma a possibilitar comunicação UART através de um *transceiver* MAX458E que está conectado aos pinos RF2 (para a transmissão da comunicação), RF8 (para a receção da comunicação), RA2 (para habilitar a transmissão) e RA3 (para habilitar a receção). O *transceiver* está conectado também a um terminal de duas entradas/saídas.

De forma a fazer uso do ADC, o PCB (*Printed Circuit Board*) contém três terminais conectados aos pinos RB13, RB14 e RB15 que correspondem aos canais analógicos AN8, AN9 e AN10 que têm “à sua disposição” módulos ADC. Entre cada terminal e pin está um circuito que contém um amplificador operacional (MCP6021T) conectado a 3,3V de forma a amplificar o sinal.

Ainda relativamente ao ADC o pin destinado à tensão de referência está conectado a uma referência de tensão (MCP1501T-25E\_RW) de 2,5V permitindo assim limitar o teto máximo das tensões recebidas pelo ADC.

A placa tem ainda conectores ligados aos pinos RG15, RA5, RE5, RE6, RE7, RC1, RC2, RC3, RC4, RG6, RG7 e RG8 de forma a serem utilizados apenas como entrada/saídas ou utilizados pelos recursos que lhes estiverem disponíveis. Na

Tabela 1 poderá ver-se os recursos disponíveis para cada pino mencionado nesta secção.

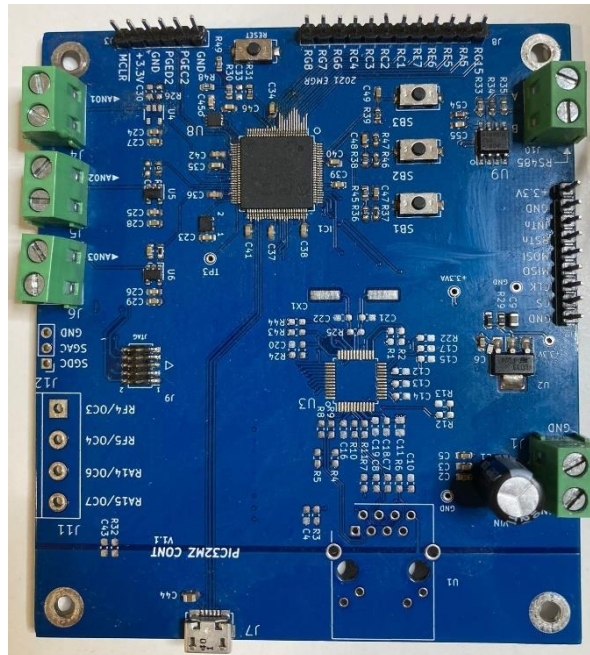


Figura 14 - Placa de Circuito Impresso Zonicstech.

Tabela 1. Função predefinida e recursos disponíveis para cada pin apresentado. (Nota: todos os pinos estão disponíveis para serem utilizados como entradas e/ou saídas assim como têm outros recursos associados que não foram mencionados na presente dissertação)

<b>Pinos</b>	<b>Função</b>	<b>Recursos</b>
RF2	Transmissão UART	U2TX e U6TX
RF8	Receção UART	U2RX
RA2	habilitação da transmissão UART	
RA3	habilitação da receção UART	
RB13	Conversão Analógico Digital	ADC7
RB14	Conversão Analógico Digital	ADC7
RB15	Conversão Analógico Digital	ADC7
RG15		
RA7		
RE5		OC4 e OC7
RE6		
RE7		
RC1		OC3, OC6 e U3TX
RC2		OC1, OC2, OC9, U2TX e U6TX
RC3		OC5, OC8, U4TX e U6TX
RC4		OC4, OC7, U1TX e U5TX
RG6		OC5, OC8, U4TX e U6TX
RG7		OC4, OC7, U1TX e U5TX
RG8		OC3, OC6 e U3TX

### **3.1.3. Motor de passo**

O motor de passo utilizado foi o 42BYGMH809, o mesmo utilizado no projeto em que esta dissertação se baseia. Este motor tem duas fases e possui um ângulo de passo de  $0,9^\circ$ , ou seja, dá 400 passos por revolução, com uma tolerância de 5%. A tensão nominal é de 2,77V e a corrente nominal por fase de 1,68A. O binário máximo é de 0,0255 Nm e o binário de retenção de 0,042 Nm (Hentai Motor, n.d.).



Figura 15 - Motor de passo 42BYGMH809 (Hentai Motor, n.d.)

### 3.1.4. *Driver do Motor*

O driver A4988SETTR-T tem como propósito controlar a rotação do motor de passo tendo resoluções de passo de 1,  $\frac{1}{2}$ ,  $\frac{1}{4}$ ,  $\frac{1}{8}$  e  $\frac{1}{16}$ . Isto é, permite que o motor, controlado por si, tenha, conforme a configuração do driver, passos até  $\frac{1}{16}$  do ângulo de passo do motor. Quanto às características, este *driver* possui uma tensão de alimentação do motor até 35V e uma corrente de saída até 2A e é compatível com tensões nível lógico de 3,3V e 5V.

O driver fornece corrente aos enrolamentos do motor e conseqüente movimento do mesmo através de duas pontes H (Figura 16). Esse fornecimento de corrente e conseqüente movimento é regulado por PWM (*Pulse-With-Modulation*) na entrada “STEP”. Na Figura 17 encontra-se o diagrama de entradas e saídas do *driver* A4988 (Allegro Microsystems, 2022). Por cada pulso que é enviado para a entrada “STEP”, o driver provoca um movimento de rotação de um passo ou um micropasso, consoante a configuração. A configuração da resolução do passo é feita através dos níveis de tensão fornecidos às entradas MS1, MS2 e MS3, conforme a Tabela 2. A entrada “DIR” tem a função de definir o sentido de rotação do motor, isto é, com a tensão a nível alto o motor irá girar no sentido dos ponteiros do relógio com a tensão a nível baixo o motor irá girar no sentido contrário ao dos ponteiros do relógio. O “RESET” restaura o *translator* para as configurações iniciais, desabilita as saídas do *driver* bem como ignora os pulsos na entrada “STEP” até ser colocado a nível alto. Já o “SLEEP” permite uma operação normal em nível alto, no entanto em nível baixo minimiza a energia consumida sendo que as saídas são desativadas bem como o regulador de corrente. Por fim, a entrada “ENABLE” serve para habilitar ou desabilitar as saídas para o motor, consoante o nível de tensão. Esta entrada, não influencia o funcionamento das entradas. Tem ainda uma entrada “VREF” que servirá para definir o valor da tensão de referência para nível alto e conseqüentemente para nível baixo (Allegro Microsystems, 2022). Na Figura 18 encontra-se apresentada a placa de circuito impresso do Driver A4988 (PTRobotics, n.d.)

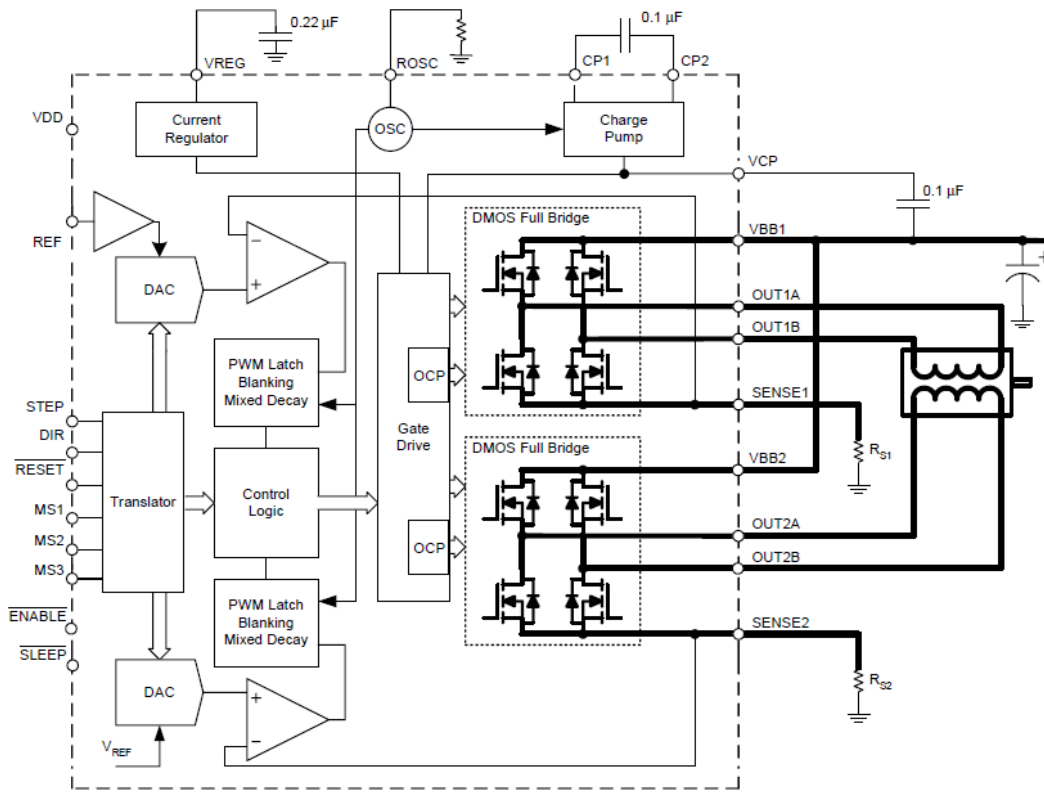


Figura 16. Diagrama Funcional do Driver A4988 (Allegro Microsystems, 2022)

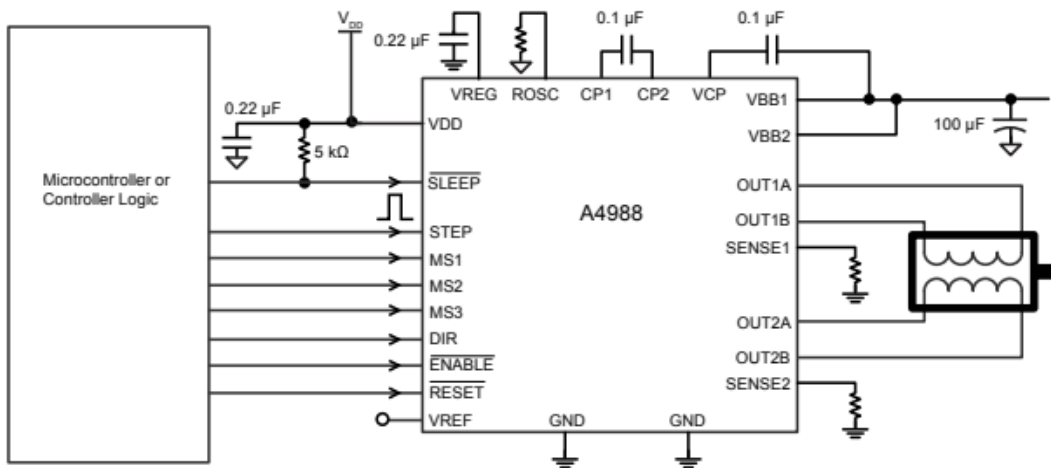


Figura 17. Diagrama de Entradas e saídas do Driver A4988 (Allegro Microsystems, 2022)



Figura 18. Placa de circuito impresso do Driver A4988 (PTRobotics, n.d.)

Tabela 2. Configuração da resolução de passo. “L” – Nível Baixo de Tensão; “H” – Nível Alto de Tensão (Allegro Microsystems, 2022)

MS1	MS2	MS3	Resolução de Passo
L	L	L	Passo completo
H	L	L	Metade de passo (1/2)
L	H	L	Quarto de passo (1/4)
H	H	L	Oitavo de passo (1/8)
H	H	H	Dezasseis avos de passo (1/16)

### 3.1.4.1. LED Infravermelho

O LED LTE-4206 (Figura 19) é um LED infravermelho que apresenta uma intensidade radiante diretamente proporcional à corrente. Admite uma corrente contínua máxima de 60mA. Apresenta uma radiância mínima de 1,383mW/sr e máxima de 7,67mW/sr. Apresenta ainda um comprimento de onda de pico de 940nm com uma corrente de 20mA (LITE-ON Technology Corporation, n.d.-a).



Figura 19. Led Infravermelho LTE-4206 (LITE-ON Technology Corporation, n.d.-a)

### 3.1.4.2. Fototransistor

O fototransistor LTR-4206E exposto na Figura 20, permite uma tensão máxima entre o coletor e o emissor de 30V e uma tensão de 5V no sentido inverso, conforme indicado na Figura 21. O conector base é ativado por irradiância apresentando uma corrente no coletor diretamente proporcional à irradiância até aos 4mW/cm<sup>2</sup>, quando a tensão entre o coletor e emissor é 5V (LITE-ON Technology Corporation, n.d.-b).



Figura 20. Foto transistor LTR-4206E (LITE-ON Technology Corporation, n.d.-b)

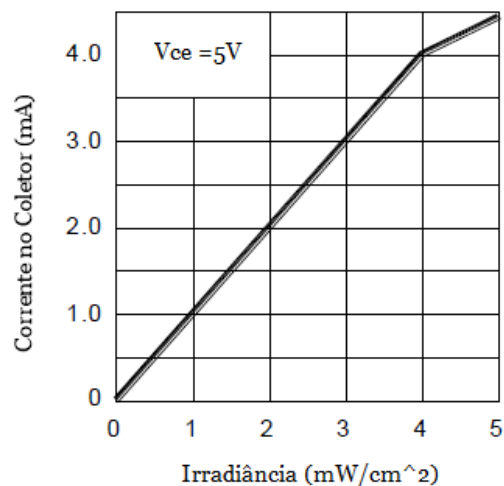


Figura 21. Relação entre corrente no coletor e irradiância na base adaptado de (LITE-ON Technology Corporation, n.d.-b)

### 3.1.5. Pickit 4

O Pickit 4 é um dispositivo desenvolvido pela Microchip Technology Inc que permita programar e fazer *debugging* de microcontroladores desenvolvidos pela mesma empresa. Este dispositivo tem uma entrada USB de forma a conectar-se com o

computador e receber a informação a programar através do IDE (*Integrated Development Environment*) e um conector de saída SIL de pinos (que permite várias interfaces) para se conectar com o microcontrolador a ser programado. Devido à capacidade de fazer *debugging* em tempo real é possível, com este dispositivo, colocar pontos de interrupção no código de forma a, por exemplo, verificar o valor de variáveis nesse momento. Este encontra-se exposto na Figura 22.

Para se efetuar um *debugging* básico do microcontrolador devem-se conectar os pinos MCLR, PGC, PGD, VDD e VSS aos mesmo respetivos pinos do microcontrolador. Os pinos PGC e PGD servem para a programação do microcontrolador e os pinos VDD e VSS para nível alto e baixo de tensão. O pin MCLR tem como propósito reinicializar o sistema quando é colocado a nível baixo e é por isso recomendado que esteja conectado a um circuito com resistência pull-up de forma a poder-se reinicializar o sistema quando desejado (Microchip Technology Inc, 2020).



Figura 22 – Pickit 4.

### **3.1.6. Conversor USB**

Na presente dissertação foi utilizado um conversor USB-Nano-485/OP. Este dispositivo converte sinais de RS-485 para USB e vice-versa. Tem uma velocidade de transmissão de 1.5Mbit/s e é alimentado através da porta USB (Ver Figura 23).



Figura 23 - Conversor USB-Nano-485

### 3.1.7. Impressão 3D

Um dos métodos utilizados para a impressão em 3D é o método por modelação por deposição fundida. Este método consiste em depositar o material a imprimir sobre a cama da impressora após ser derretido por uma extrusora. O material é depositado por camadas. Isto é, a extrusora primeiro deposita o material uma camada ao longo dos dois eixos horizontais e posteriormente deposita outras camadas sobre as camadas já depositadas até a peça a imprimir estar concluída (Solomon *et al.*, 2020).

Um dos materiais mais utilizados na impressão 3D é o PLA (ácido polilático). Este tipo de material trata-se de um termoplástico constituído por material biodegradável com uma temperatura de fusão a rondar os 170°C que ao arrefecer volta ao estado sólido. Algumas das razões pelas quais é um dos materiais mais utilizados na impressão 3D são a facilidade na impressão, qualidade na resolução da impressão e ser amigo do ambiente (ALL3DP & Jackson O'Connel, n.d.).

Para o processo de impressão é necessário ter o modelo 3D da peça que se pretende imprimir. Este modelo deve ser convertido num ficheiro STL que através de um *software* CAM (*Computer Aided Manufacturing*), é possível definir as configurações de impressão como a velocidade de impressão, a temperatura da extrusora e da cama da impressora, espessura das paredes e densidade da peça, ou seja, é possível obter peças completamente ocas (só com a espessura das paredes) bem como peças completamente maciças. Após estas configurações estarem definidas, deve-se exportar, através do *software* CAM, um ficheiro de código G. Será através deste ficheiro de código G que a impressão será feita. Após a impressora terminar a impressão deve-se remover a peça da cama que em muitos dos casos será necessário retirar o suporte, limpar fios e/ou fazer outros tratamentos à peça como lixar e polir (My3DConcepts, n.d.).

### 3.1.7.1. Artillery Sidewinder X1

A impressora utilizada na presente dissertação foi a impressora *Artillery Sidewinder X1* exposta na Figura 24. Esta impressora possui uma resolução de impressão de 0,05mm nos eixos horizontais e de 0,1mm no eixo vertical. A extrusora da mesma possui um bocal de 0,4mm o que fará com que o filamento extraído seja sensivelmente do mesmo diâmetro. A impressora possui ainda um *display touch screen* que permite alterar algumas configurações da impressora bem como escolher o ficheiro STL da peça que se pretende imprimir (3D Printer Universe, n.d.).



Figura 24 – Impressora 3D *Artillery Slidewinder X1* (3D Printer Universe, n.d.)

### 3.1.8. Soldadura Manual

A soldadura manual em placas de circuito impresso consiste num processo que tem como objetivo fixar componentes a um condutor. De forma a fixar os componentes é utilizado um fio de estanho ou pasta de soldadura que são derretidos para que os componentes fiquem fixos. Para derreter a pasta de soldadura é utilizado um ventilador de ar quente enquanto para derreter o fio de estanho é utilizado encostado o ferro, da estação de soldadura, que também se encontra a elevadas temperaturas. Após a pasta ou pedaço de estanho arrefecer volta a solidificar fixando o componente que se pretendia (Precision PCB Services, n.d.). Na Figura 25 é exposta a estação de soldadura com ferro e secador.



Figura 25 - Estação de Soldadura com ferro e secador.

### 3.1.9. Matlab

O Matlab trata-se de um *software* utilizado como ferramenta de programação, análise e simulação numérica. Alguns dos campos onde o Matlab mais é usado são *machine* e *deep learning*, processamento de sinal, robótica, etc.. Por isso mesmo, é um *software* bastante utilizado por engenheiros e investigadores (Reis *et al.*, 2022). Uma das grandes mais valias do Matlab é a capacidade e facilidade que o *software* tem em manipular matrizes. Com grande facilidade permite o utilizador visualizar uma matriz de dados num gráfico.

Possui ainda uma grande biblioteca de funções que facilitam o trabalho ao utilizador que não tem necessidade de implementar funções de baixo nível para efetuar os cálculos ou processamento de dados que pretende. Possui funções simples que permitem extrair a soma ou a média de todos os elementos (Asadi, 2023) de uma matriz, mas possui também funções mais complexas como o cálculo da transformada rápida de Fourier (MathWorks, n.d.-c) e a aplicação de um filtro digital a uma amostra de dados (MathWorks, n.d.-a).

Para além disto possui ainda inúmeras ferramentas para auxiliar processos. Uma das quais para dimensionar coeficientes de filtros digitais, chamada *Filter Designer*. Esta ferramenta permite dimensionar os coeficientes de um filtro digital escolhendo o tipo de resposta, o tipo de filtro, o número de coeficientes e a frequência de corte (MathWorks, n.d.-b).

### 3.1.10. Mplab X IDE

O Mplab X IDE é um compilador que permite desenvolver e fazer *debug* de código para microcontroladores da Microchip. Este programa suporta as linguagens de programação C, C++ e Assembly.

Entre outras características, este compilador permite ao utilizador colocar *breakpoints* em determinadas zonas do código para que o código pare de correr nesse ponto quando se está a fazer o *debugging*. Quando o processamento para no *breakpoint*, o compilador permite observar o valor das variáveis que se desejam nesse momento.

O compilador permite ainda saber previamente ao *debugging* qual a percentagem da memória que as variáveis criadas e o restante código irão ocupar no microcontrolador a utilizar (Microchip Technology Inc, 2022). Na Figura 26 é exposta a janela de interface do Mplab X IDE.

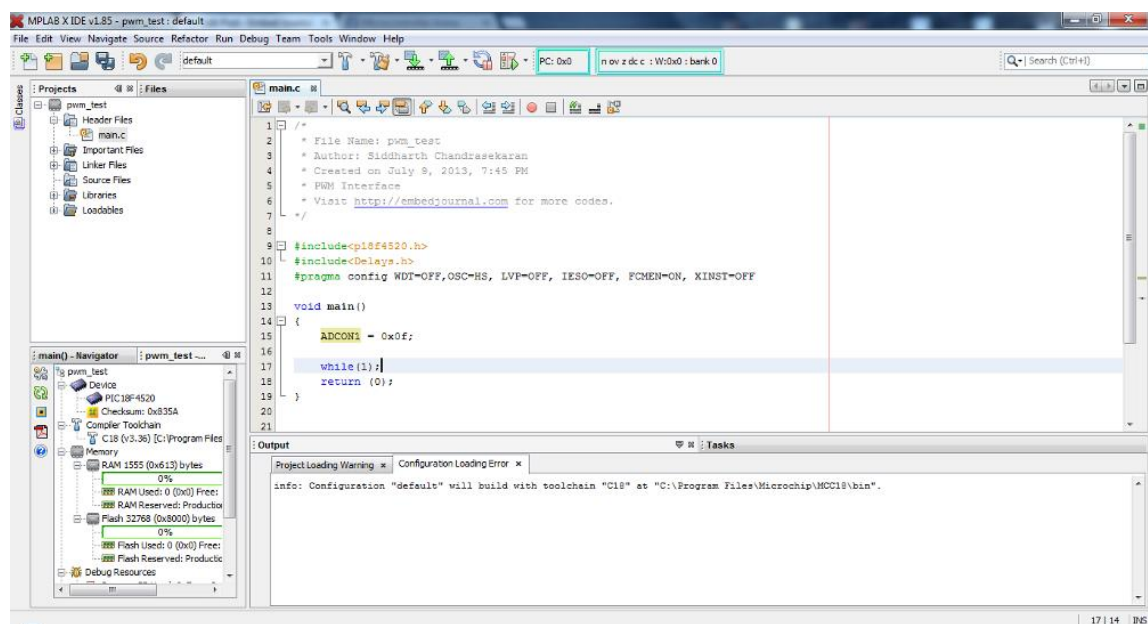


Figura 26 – Interface do *Mplab X IDE* (Microchip Technology Inc, 2022).

### 3.1.11. SerialPlot

SerialPlot trata-se de um *software* que permite receber dados via comunicação UART no computador através de uma porta USB. Este programa permite receber dados de 8, 16, 32, float, double e ASCII e exportá-los para um ficheiro CVS. O programa permite ainda

observar num gráfico o valor dos dados recebidos em tempo real Özderya, n.d.). Na Figura 27 é apresentado o interface do SerialPlot.

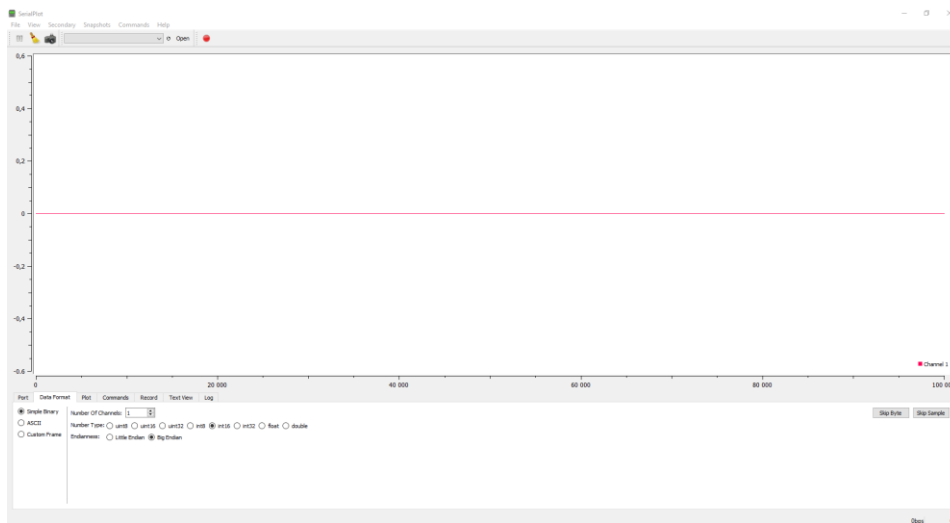


Figura 27 – Interface do SerialPlot.

### 3.1.12. Kicad

O Kicad é um *software* de EDA (EDA – *Electronics Design Automation*). Este *software* está contêm um editor esquemático, um editor de símbolos, um editor de layout de placas de circuito impresso, um editor de *footprints* e um visualizador 3D.

O editor esquemático permite ao utilizador desenhar um circuito eletrónico com os componentes disponíveis nas bibliotecas do *software*. Este editor permite criar várias páginas de forma hierárquica para que no caso de desenhos eletrónicos complexos, o utilizador possa organizar o *layout* esquemático como lhe agradar. O editor de símbolos permite criar novas bibliotecas e símbolos para poderem ser utilizados no editor esquemático. Já o editor de *footprints* permite que o utilizador associe *footprints*, já existentes nas bibliotecas deste editor, a componentes. Este editor permite também que o utilizador crie *footprints* caso as medidas que pretende não estejam na biblioteca. Ao ter-se todos os componentes do desenho eletrónico no editor esquemático associados a *footprints*, é possível desenhar uma placa de circio impresso através do editor de *layout* da mesma. O visualizador 3D permite importar modelos 3D dos componentes e associá-los aos mesmos de forma a que o utilizador possa visualizar um modelo 3D do *layout* da placa antes de mandá-la para fabrico (Kicad EDA, 2021).

### **3.1.13. Solidworks**

Atualmente, a criação de qualquer sistema mecânico passa pela modelação tridimensional. Esta é feita através de *softwares* de desenho assistido por computador (CAD – *Computer Aided Design*).

Um dos *softwares* mais utilizados na indústria é o *SolidWorks*. Este *software* permite a criação de objetos 3D complexos a partir de formas geométricas simples. Após as peças estarem completas, o *SolidWorks* permite a sua modificação de forma simples o que o torna bastante versátil. Este tem ainda outras funcionalidades como apresentação dos planos das peças 3D em 2D e respetivas medidas e ferramentas de simulação elétrica, aerodinâmica e esforço mecânico (Solidworks, 2015).

## 4. Protótipo

### 4.1. Desenvolvimento do codificador ótico

O propósito do codificador ótico é produzir duas ondas sinusoidais com um defasamento de  $90^\circ$  elétricos, à medida que é conferido um movimento giratório à estrutura, para posteriormente ser calculado o ângulo do movimento através de um algoritmo no microcontrolador. Para este efeito são necessários tanto componentes mecânicos como componentes eletrônicos.

#### 4.1.1. Desenho eletrônico

O desenho eletrônico da placa de circuito impresso do codificador ótico foi baseado no desenho eletrônico do projeto *Dexter 1* da *Haddington Dynamics* (Kochis & Haddington Dynamics, 2015).

Como se poderá visualizar na Figura 28, o esquema eletrônico do codificador ótico é composto por 2 circuitos com um LED e um fototransistor cada, ambos alimentados com 3,3V. Os LEDs estão protegidos por uma resistência de  $100\ \Omega$  e um potenciômetro com uma resistência máxima de  $2k\Omega$ . A corrente que é fornecida aos LEDs pode ser dada pelas Equações 22 e 23. Uma vez que ambos os circuitos possuem a resistência de  $100\ \Omega$  a corrente que passa pelo LED nunca será superior  $33mA$ , cumprindo o requisito de corrente máxima de  $60mA$ . O potenciômetro tem como propósito regular a intensidade radiante que chega aos fototransistores de forma que estes produzam sinais com amplitudes semelhantes. O emissor dos fototransistores está ligado a um terminal de forma a ligar-se ao ADC (Figura 28 e Figura 29). O propósito das resistências colocadas entre a ligação com o terminal e a referência de potencial zero é criar um divisor de tensão, dado que o fototransistor funcionará como uma resistência que variará inversamente proporcional à intensidade radiante recebida. Por outras palavras, quanto maior a radiância recebida pelo fototransistor maior será a corrente no coletor e por consequência no emissor, logo, menor será a resistência à passagem de corrente. O objetivo será criar uma radiância sinusoidal na base do fototransistor de forma que este permita uma passagem de corrente sinusoidal para que seja transformada numa tensão sinusoidal pelo “divisor de tensão”.

$$I_{led1} = \frac{V}{R1+R2} \quad (22)$$

$$I_{led2} = \frac{V}{R4+R5} \quad (23)$$

$$V_{R3} = V * \frac{R3}{R3+R_{Fototransistor1}} \quad (24)$$

$$V_{R6} = V * \frac{R6}{R6+R_{Fototransistor2}} \quad (25)$$

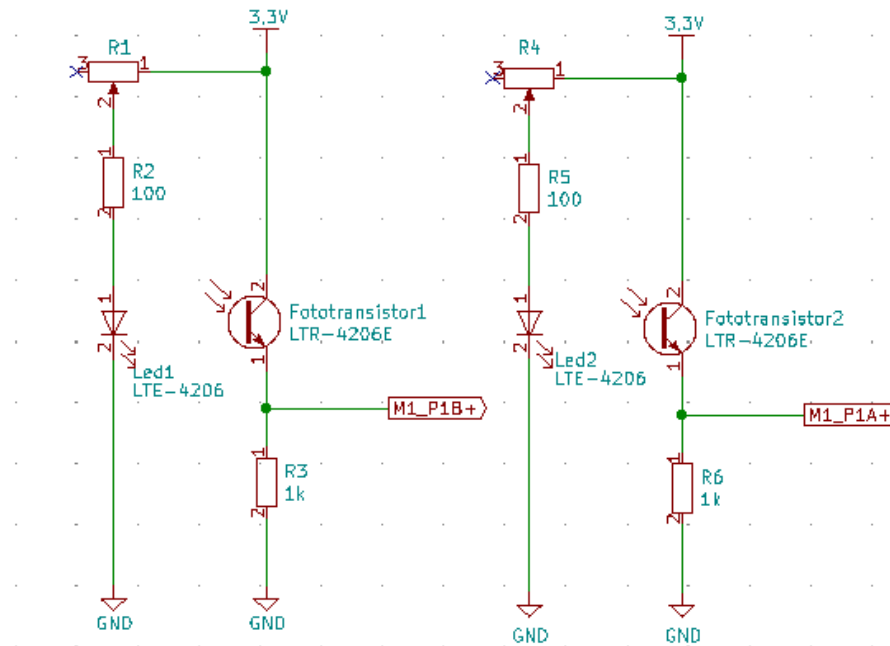


Figura 28. Esquema elétrico dos circuitos led – fototransistor.

A alimentação dos circuitos LED-fototransistor é feita através do terminal J1 (Figura 29). Alimentação essa que é primeiramente filtrada de forma a adicionar o mínimo de ruído nos circuitos LED-fototransistor.

As *ferrite beads* no início do circuito têm como propósito filtrar o ruído de alta frequência. Este componente age como uma resistência dissipando o ruído de alta frequência como calor, durante a zona de resistividade que corresponde desde, sensivelmente, os 10MHz até perto dos 1000MHz (Figura 30). A indutância até aos 10MHz pode ser calculada através da Equação 26 e a capacidade posterior a 1000MHz pode ser calculada através da Equação 27. Quer isto dizer que as *ferrite beads* do circuito irão filtrar ruído entre 10MHz e 1000MHz (Eco & Limjoco, 2016).

$$L = \frac{X_L}{2 * \pi * f} \quad (26)$$

$$C = \frac{1}{2 * \pi * f * |X_C|} \quad (27)$$

De seguida é utilizado um filtro de pi rejeita banda de forma a atenuar-se a amplitude de determinadas frequências

De modo geral, o circuito pretende estabilizar a tensão de alimentação dos LEDs e fototransístores e minimizar o ruído para que não interfira com o sistema de posicionamento.

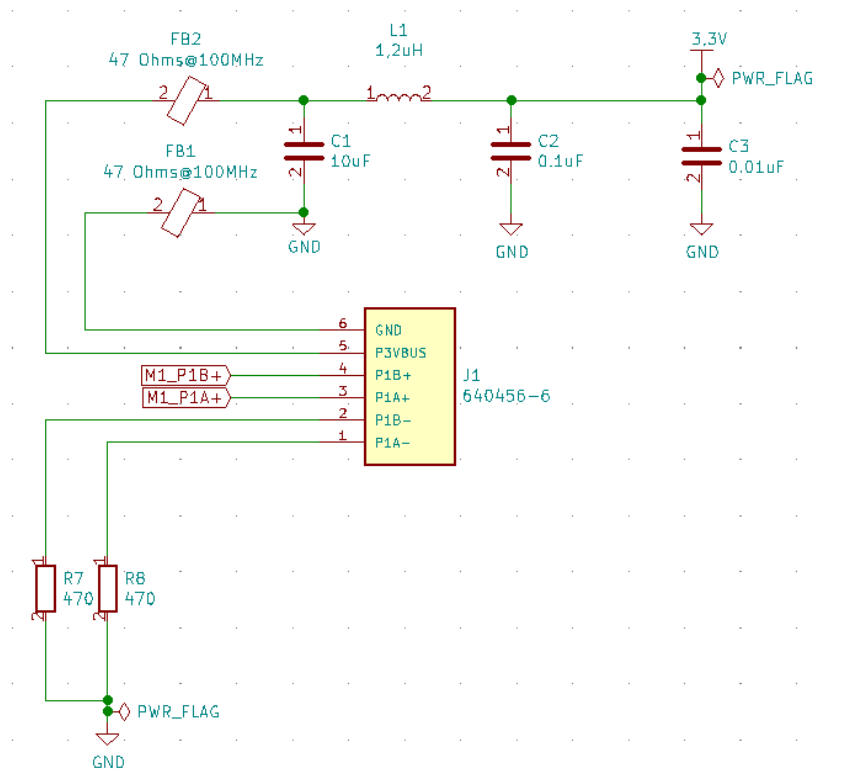


Figura 29. Desenho elétrico do terminal que fornece a tensão de alimentação e recolhe a tensão à saída do fototransísttor.

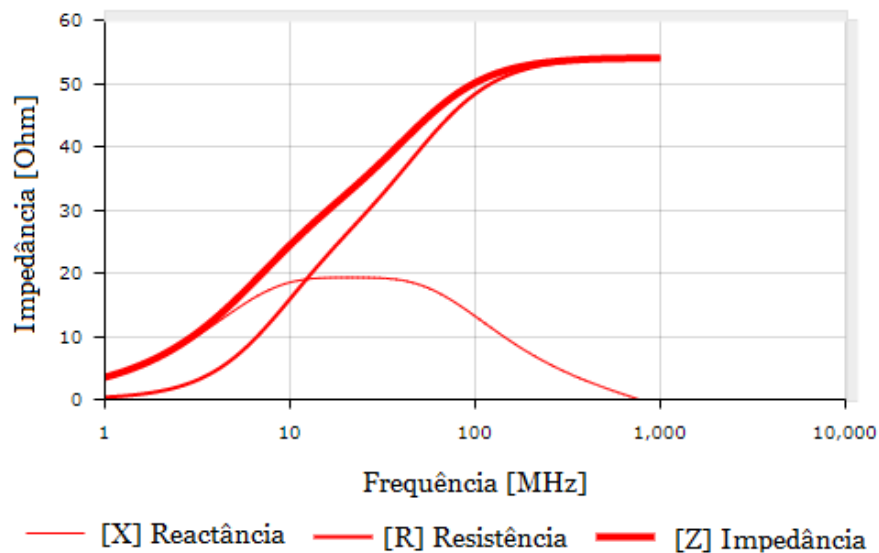


Figura 30 - Gráfico de impedância, resistência e reactância da *ferrite bead* em função da frequência.  
Adaptado de (Yuden, n.d.)

Posteriormente ao desenho elétrico, foi feito o desenho da placa de circuito impresso para posteriormente se soldarem os componentes eletrônicos à mesma e ser utilizada no projeto (Figura 31). Os *footprints* onde estão as ligações para os fototransistores e LEDs não correspondem aos *footprints* dos mesmos, mas sim de terminais que por sua vez se conectaram a estes. O *footprint* no canto inferior esquerdo corresponde ao terminal J1 (Figura 29) que fornece a tensão de alimentação e recolhe os dados dos fototransistores. Os furos H1 e H2 distanciados 24 mm servirão para encaixar na estrutura mecânica. O resultado final do circuito impresso já com os componentes soldados está representado na Figura 33.

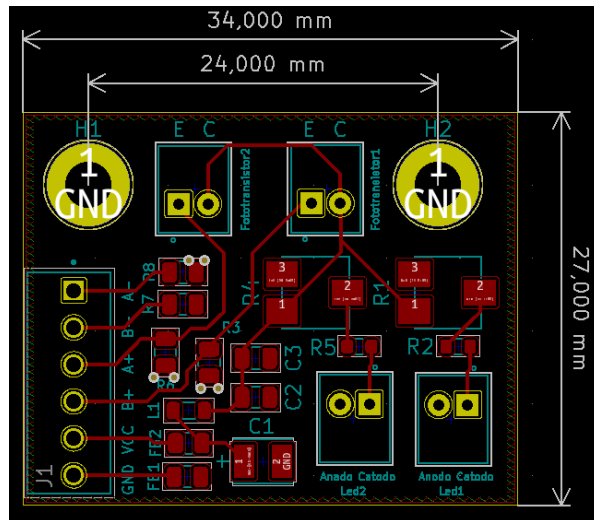


Figura 31. Desenho da placa de circuito impresso

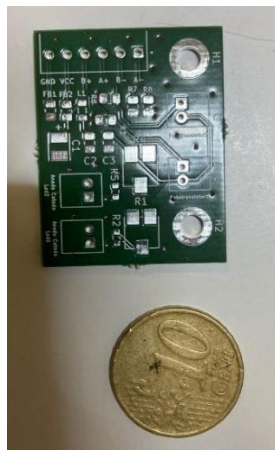


Figura 32 - Circuito impresso já com componentes por soldar

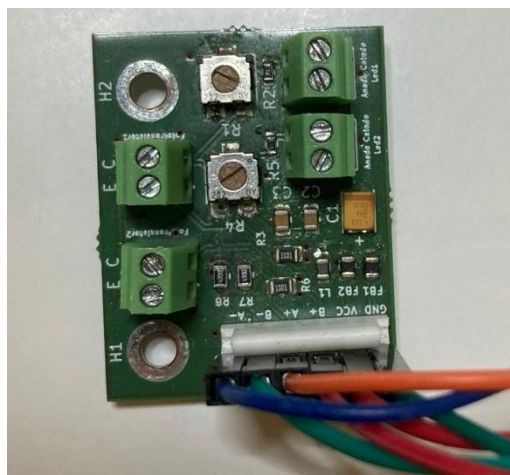


Figura 33 - Circuito impresso já com componentes soldados.

#### 4.1.2. Componente mecânica

O objetivo da componente mecânica do codificador ótico é fazer com que o transístor receba valores de radiância sinusoidais à medida que é conferido movimento à estrutura. Para isso, esta é constituída por 3 peças impressas em PLA: Interruptor de Radiância (Figura 34 e Figura 35), Disco Codificador (Figura 36) e Cobertura do Codificador (Figura 37).

O Interruptor de Radiância tem como propósito alinhar os dois LEDs com os dois fototransístores através dos 4 furos circulares que possui. Como se poderá verificar nas figuras, os furos estão alinhados em pares. Isto serve para que os LEDs e fototransístores estejam também alinhados. Para possibilitar a passagem de radiação do LED para o fototransístor a peça possui um furo retangular, transversal e coincidente com cada par de furos.

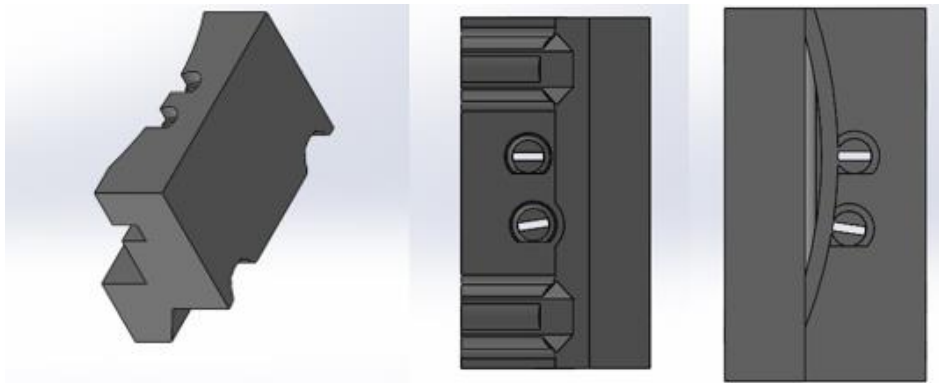


Figura 34 - Várias vistas do Interruptor de Radiância.

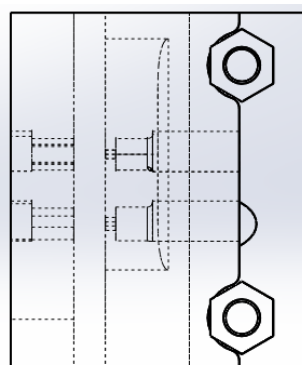


Figura 35 - Vista superior dos furos do Interruptor de Radiância.

O Disco Codificador (Figura 36) é um disco com 200 furos distribuídos uniformemente ao longo da circunferência que coincidirão com os furos do Interruptor de Radiância. Sendo os furos do disco coincidentes com os furos do Interruptor de Radiância com que o movimento circular provoque uma radiância sinusoidal na base dos fototransístores.

Devido ao tamanho dos LEDs e fototransístores, torna-se impossível que dois pares LED/fototransistor estejam sobre o mesmo furo, o que obriga a que estejam a ler dados sobre furos diferentes. Dado que os ciclos sinusoidais se repetem a cada  $1,8^\circ$  mecânicos ( $360^\circ/n^\circ$  de furos), é possível ter os dois pares LED-fototransístores sobre furos diferentes desde que mantenham o desfasamento de  $90^\circ$  elétricos. Idealmente os furos dos LED-fototransístores estariam distanciados de  $0,45^\circ$  mecânicos ( $90 \cdot 1,8/360$ ). Neste caso estão distanciados de  $8,55^\circ$  mecânicos que corresponde a um desfasamento de  $270^\circ$  ou  $-90^\circ$  elétricos (na prática).

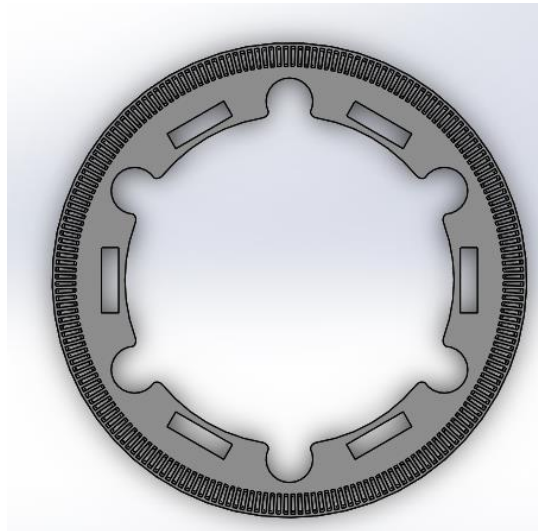


Figura 36 – Disco Codificador

A cobertura do codificador (Figura 37) tem como função cobrir o Interruptor de Radiância de modo a permitir que o mínimo de luminosidade interfira nos fototransístores. Esta peça é também onde encaixará a placa de circuito impresso como se verá no próximo capítulo.

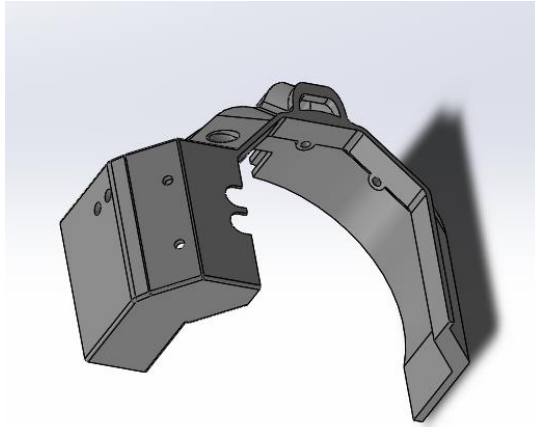


Figura 37 – Cobertura do Codificador

#### 4.1.3. Montagem da estrutura

A estrutura mecânica montada foi baseada na estrutura do braço robótico “Dexter 1” da *Haddington Dynamics* (Newton & Haddington Dynamics, 2020). Todas as peças foram impressas em PLA. Algumas delas foram impressas de acordo com o modelo original, outras sofreram algumas alterações no desenho.

Os materiais utilizados estão representados na Tabela 3 e Tabela 4. A estes materiais acrescem ainda parafusos e porcas M3, uma abraçadeira de alumínio e o motor de passo.

Tabela 3 – Peças Impressas em 3D

<b>Peças de PLA</b>	
Descrição	Quantidade
Disco Codificador	1
Coluna da Base	1
Base de montagem inferior	1
Base de montagem superior	1
Interruptor de radiância	1
Pivô	1
Suporte superior do motor	1
Suporte inferior do motor	1
Cobertura do Codificador	1

Tabela 4 - Materiais para a construção da estrutura

Descrição	Quantidade	Comprimento [mm]	Largura [mm]	Espessura [mm]
Barras de Carbono	6	135	12,6	3
Barras de Carbono	3	126	12,6	3
Descrição	Quantidade	Diâmetro Interno [mm]	Diâmetro Externo [mm]	Espessura [mm]
Rolamentos	2	50	65	7

O primeiro passo na montagem da estrutura foi o encaixe dos suportes inferior e superior do motor (Figura 38). De seguida, encaixa-se um dos rolamentos na coluna da base, onde posteriormente se encaixa o motor com os suportes (Figura 39 e Figura 40). Na outra extremidade da coluna da base é encaixado o outro rolamento no suporte inferior do motor (Figura 41). Os rolamentos permitirão ao motor rodar mantendo a coluna da base fixa.

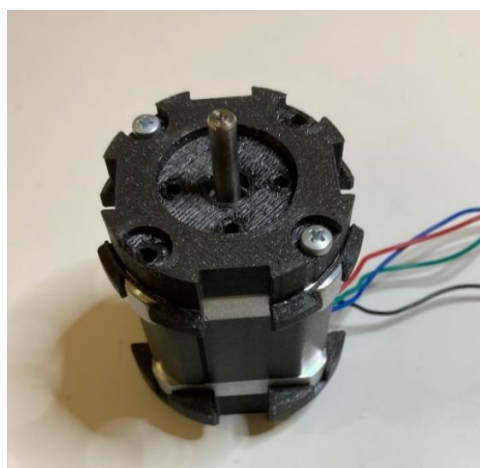


Figura 38 - Motor com os suportes inferior e superior.



Figura 39 - Coluna da base com um rolamento.



Figura 40 – Coluna da base com um rolamento e motor encaixados.

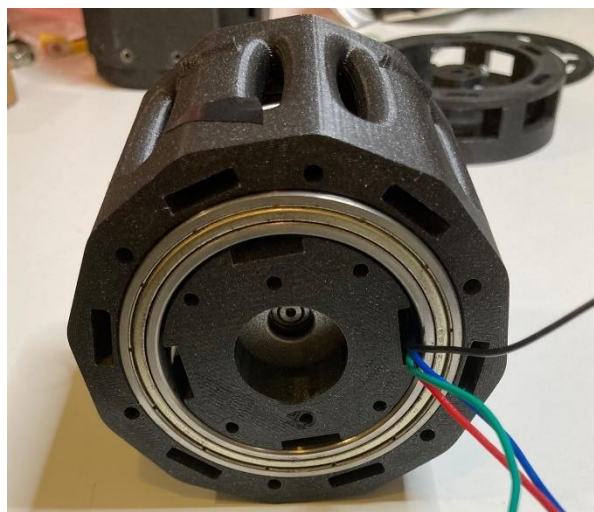


Figura 41 – Coluna da base com ambos os rolamentos e motor encaixados.

Nesta estrutura será o motor a rodar enquanto o veio estará fixo. Para isso é necessário que o veio exerça binário sobre outra peça. Com esse intuito foi desenhada a base de montagem superior, onde encaixa o veio e é fixo por dois parafusos dispostos com 90° entre si (Figura 42). Após se encaixar o veio no furo da base de montagem superior devem-se apertar bem os dois parafusos de forma a que o veio fique bem fixo e não escorregue (Figura 43).



Figura 42 – Base de montagem superior e coluna da base com o motor.

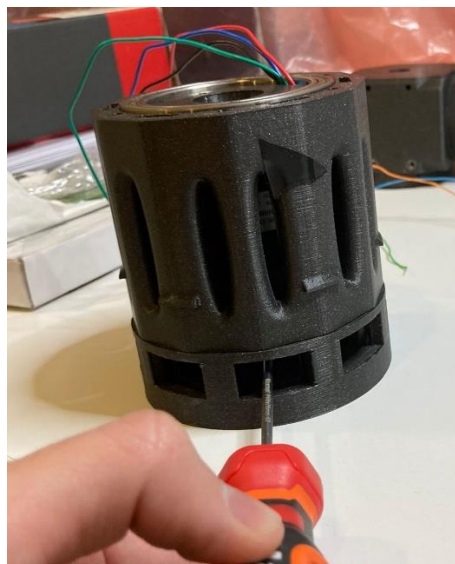


Figura 43 – Aperto dos parafusos da base de montagem superior.

Os furos ao longo do perímetro das duas peças servirão para colocar as 6 barras de carbono de 135mm de comprimento e fixá-las juntamente com a base de montagem inferior. As barras de carbono devem ser empurradas o máximo para baixo e de seguida deve-se apertar a braçadeira para que fiquem bem fixas. Na Figura 45 as barras ainda não se encontram totalmente empurradas para baixo, que nesta posição impediriam a rotação do pivô e conseqüentemente do motor.



Figura 44 – Base de Montagem inferior com as 6 barras de carbono encaixadas.



Figura 45 – Base de Montagem inferior já montada na estrutura.

O disco codificador encaixa nas barras de carbono.

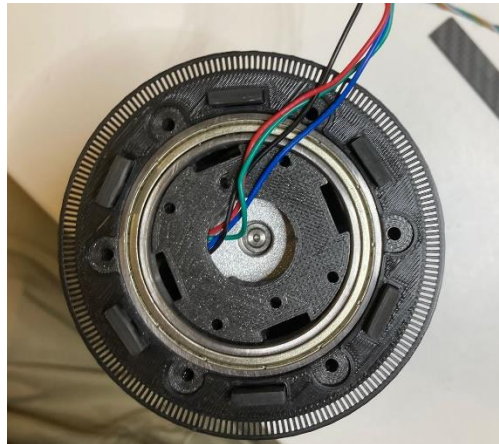


Figura 46 - Disco Codificador na estrutura.

Os furos visíveis no suporte do motor inferior servem para as 3 barras de carbono restantes. Estas 3 barras de carbono encaixam nos dois suportes do motor, já que na outra extremidade o suporte do motor superior tem cavidades idênticas e no pivô. Isto faz com que o pivô e peças acopladas ao mesmo rodem solidariamente com o motor.

De seguida foi apertado o interruptor de radiância ao pivô através dos furos que este tem onde previamente foram coladas duas porcas M3. Também nos furos do interruptor de radiância foram coladas duas porcas M3 para permitir o aperto, bem como colocados os LEDs e fototransístores nos furos que lhes pertencem. O aperto desta peça ao pivô deve ser cuidadosa de modo a que fique o mais encostada e paralela ao pivô. Esta necessidade deve-se ao facto da ranhura do interruptor de radiância coincidir com o disco codificador. Quanto mais encostada e paralela esta peça estiver ao pivô melhor alinhamento terão os pares LED-fototransistor com os furos do disco.

Os últimos passos da montagem da estrutura passam por colocar a cobertura do codificador e apertá-la à estrutura através de mais dois furos com porcas coladas que o pivô tem (Figura 51), e apertar a placa de circuito impresso à cobertura bem como ligar os LEDs e fototransístores aos terminais da placa (Figura 52).

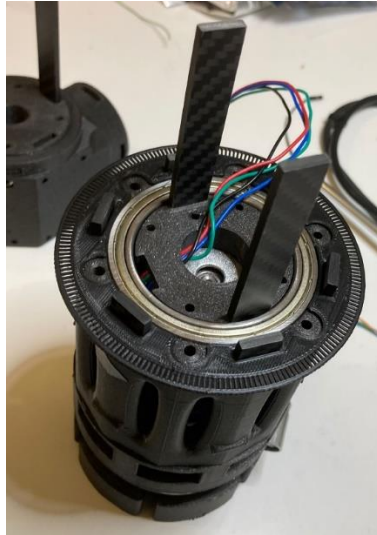


Figura 47 – Estrutura com duas barras de carbono nos suportes do motor.



Figura 48 – Pivô já encaixado na estrutura.

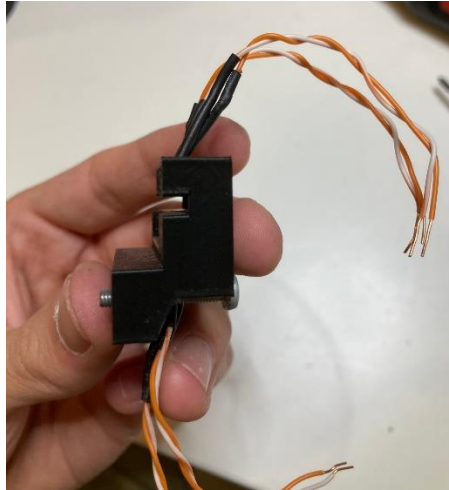


Figura 49 – Interruptor de radiação.



Figura 50 – Interruptor de radiação na estrutura.



Figura 51 - Cobertura do codificador apertada à estrutura.

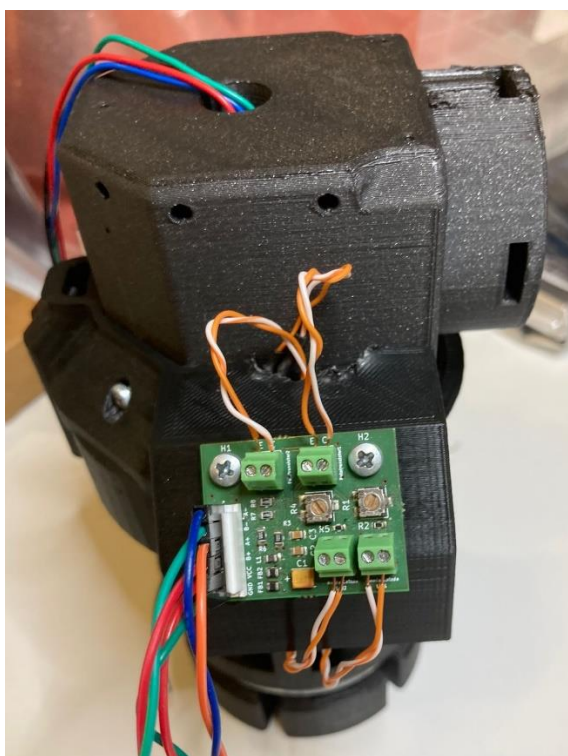


Figura 52 - Placa de circuito impresso montada na estrutura.

#### 4.1.4. Conexões

O passo seguinte passa por estabelecer as ligações necessárias de forma a testar o circuito e retirar dados para analisar. Essas ligações estão representadas na Figura 53, em que as linhas representam ligações com a fonte de tensão e as setas representam comunicação e o sentido da mesma.

Para se estabelecerem as conexões foi colocada a placa de circuito impresso, disponibilizada pela Zonicstech, na estrutura. Para o filtro RC e para o *driver* do motor foi utilizada uma *breadboard* que foi colocada no topo da estrutura.

Na Tabela 5 estão representados os pinos do microcontrolador utilizados para fazer essas ligações.

Começando pelo *driver* que está naturalmente ligado ao motor que confere movimento à estrutura. A alimentação do mesmo é feita por 24V, tensão que fica dentro dos limites máximos e mínimos (8V e 35V) especificados na ficha técnica do *driver*. A tensão de referência escolhida é 3,3V, uma vez que é o valor de tensão da saída dos pinos do microcontrolador para nível alto. Os pinos !SLEEP e !RESET não estão representados porque é sugerido, para um funcionamento normal, que sejam circuitados um com o outro.

A tensão à saída dos fototransístores passa primeiro por um circuito de filtro de passa-baixo de forma a eliminar o ruído de alta frequência antes de ser convertido em sinal digital pelo conversor analógico digital.

Após os dados serem convertidos em sinal digital pelo conversor analógico digital é necessário analisá-los. Graças ao *transceiver*, é possível enviar os dados para um computador através de comunicação UART. O *software SerialPlot* tratará de traduzir a comunicação recebida pelo computador.

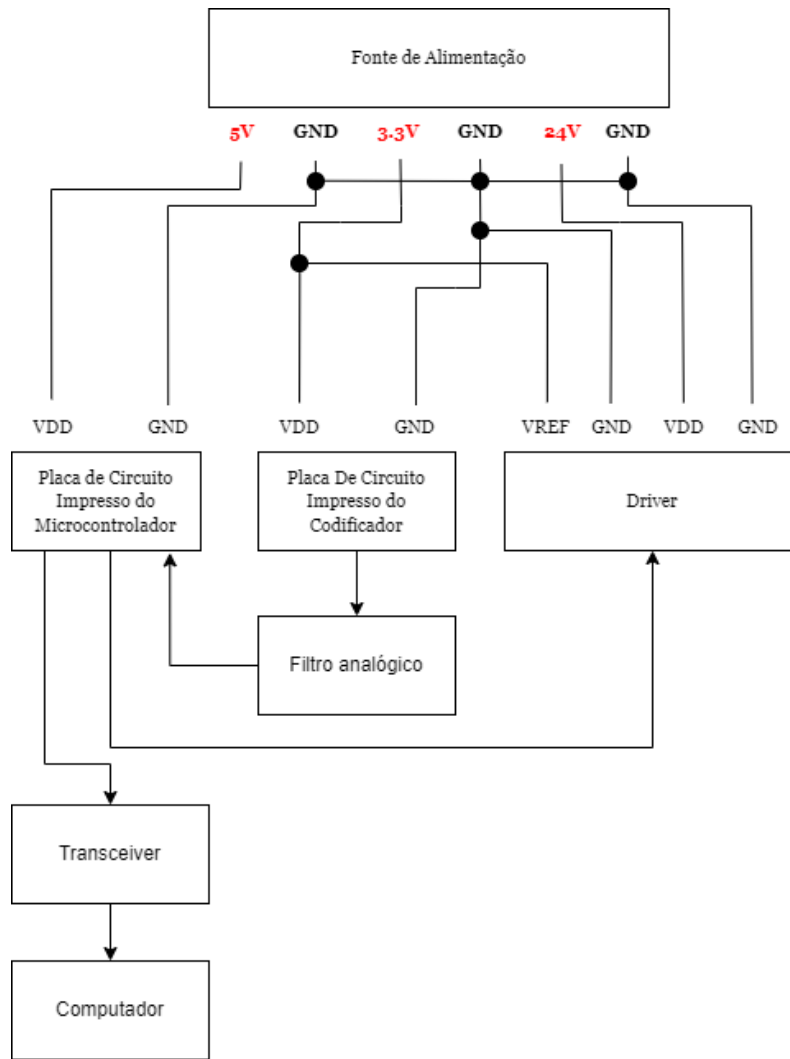


Figura 53. Ligações entre os diferentes dispositivos.

Tabela 5. Ligação entre pinos do microcontrolador e outros dispositivos.

<b>Microcontrolador</b>	<b>Driver</b>	<b>Funcionalidade</b>
RG6	MS3	Saída
RG7	MS2	Saída
RG8	MS1	Saída
RG15	!ENABLE	Saída
RE7	DIR	Saída
RC1	STEP	Comparador de saída
<b>Microcontrolador</b>	<b>PCB Encoder</b>	<b>Funcionalidade</b>
RB14	Fototransistor1	ADC
RB15	Fototransistor2	ADC
<b>Microcontrolador</b>	<b>Transceiver</b>	<b>Funcionalidade</b>
RF2	TX	Transmissão UART
RF8	RX	Receção UART
RA2	ENABLE TX	Saída
RA3	ENABLE RX	Saída

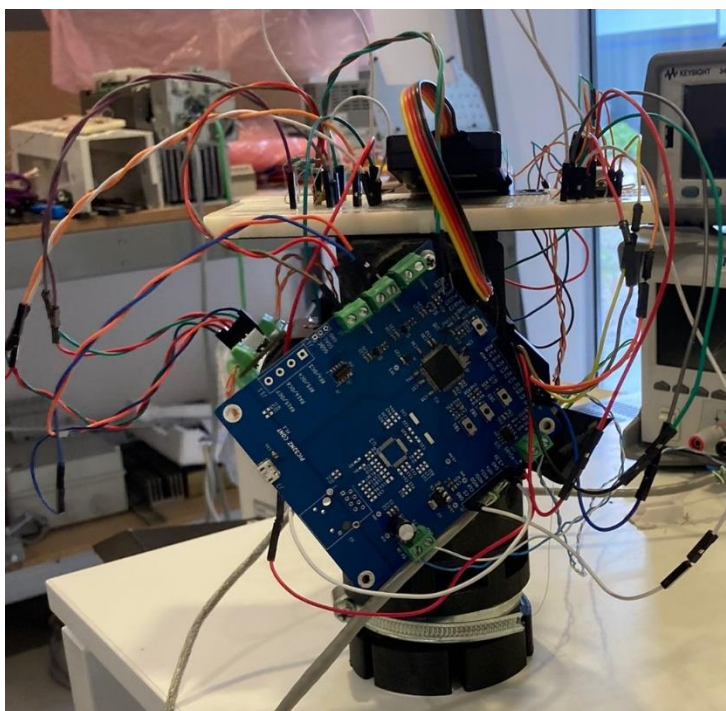


Figura 54 – Estrutura montada com a placa da Zonicstech e *breadboard*.

#### 4.1.5. Filtro Analógico

De forma a reduzir a magnitude de possíveis ruídos é necessário dimensionar filtros analógicos e digitais. Para o primeiro teste foi dimensionado apenas um filtro analógico deixando-se o dimensionamento do filtro digital para futuros testes.

Uma vez definida a frequência do ciclo elétrico como 20Hz, o objetivo do filtro analógico será atenuar a magnitude de frequências mais elevadas que esta. Ou seja, é dimensionado um filtro passa-baixo. Quanto menor for a frequência de corte deste filtro, maior será a amplitude de frequências que será capaz de reduzir e maior será a atenuação em altas frequências. No entanto, quanto mais próximo for a frequência de corte do valor da frequência do harmônico principal (20Hz), maior será o desfasamento causado a este sinal. Uma vez que o desfasamento do sinal deve ser o menor possível, para uma primeira iteração, estipulou-se que o desfasamento causado pelos filtros analógico e digital não deve ultrapassar 1° (elétrico). Foi estipulado serem criadas duas condições com pesos diferentes no desfasamento causado por cada filtro. Isto é, para a condição A o filtro analógico e o filtro digital causam sensivelmente o mesmo desfasamento ao sinal (idealmente 0,5° + 0,5°), e para a condição B o filtro analógico causa menor desfasamento que o filtro digital.

Para a definição da frequência de corte da condição A, definiu-se os valores da resistência e condensador através uma dedução da Equação 11, obtendo-se a equação 29:

$$R * C = \frac{\tan(\alpha)}{2 * \pi * f} \quad (28)$$

Onde  $\alpha$  representa o desfasamento e foi atribuído 0,5° e  $f$  representa a frequência do harmônico principal (20Hz). Traduzindo-se assim a frequência de corte ideal como 2918Hz através da equação 30:

$$f_c = \frac{1}{2 * \pi * R * C} \quad (29)$$

Uma vez que os valores mais próximos de condensadores e resistências disponíveis eram 0,8uF e 220Ω, utilizaram-se o condensador desse valor e 3 resistências em paralelo de 220Ω, perfazendo uma resistência equivalente de 86,6Ω, resultando numa frequência de corte de 2957Hz e um desfasamento de -0,4991°.

Para a condição B utilizou-se o mesmo valor para o condensador e utilizaram-se 4 resistências de 220 Ω em paralelo, perfazendo um resistência equivalente de 55 Ω, resultando numa frequência de corte de 3617Hz e um desfasamento de -0,3168°.

Os diagramas de Bode dos filtros estão representados na Figura 55 e na Figura 56.

Nota: foram utilizados dois filtros um para cada sinal, para cada condição.

Condição A:

- Frequência de corte = 2957Hz
- Resistência Equivalente = 86,6  $\Omega$
- Condensador = 0,8uF
- Desfasamento = -0,4991°

Condição B:

- Frequência de corte = 3617Hz
- Resistência Equivalente = 55  $\Omega$
- Condensador = 0,8uF
- Desfasamento = -0,3168°

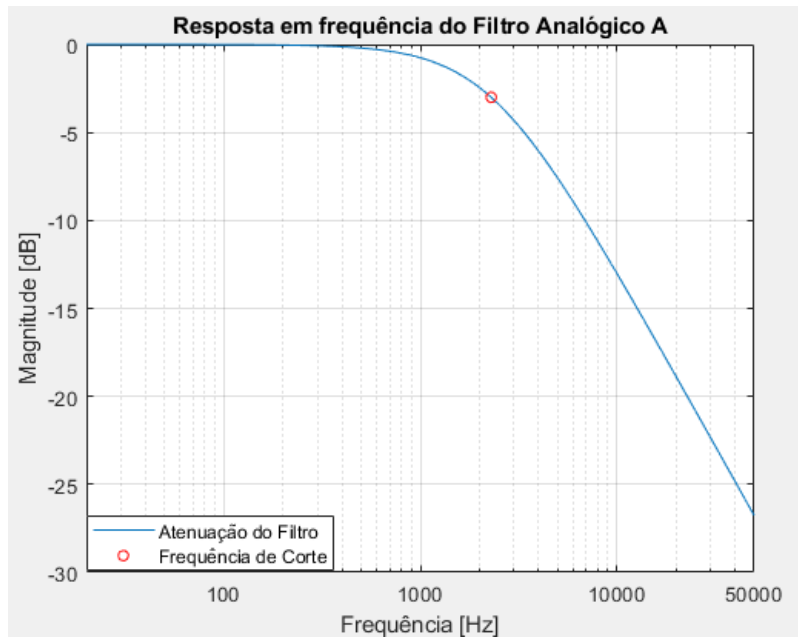


Figura 55 – Diagrama de bode do Filtro da Condição A.

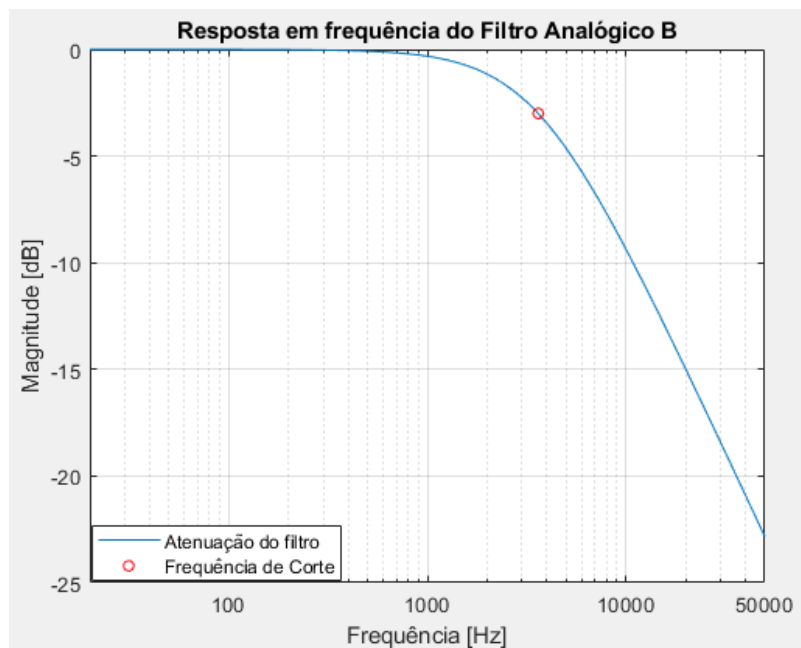


Figura 56 - Diagrama de bode do Filtro da Condição B.



## 5. Análise e Discussão de Resultados

### 5.1. Primeiro Teste

#### 5.1.1. Preparação do primeiro teste

O objetivo do 1º teste passa por retirar uma amostra representativa das ondas sinusoidais dos fototransistores e analisar as mesmas. Para tal, é necessário conferir rotação ao motor através da configuração do comparador de saída, configurar o conversor analógico digital para recolher os dados e configurar a UART para enviar os dados para o computador. Estas configurações vão depender das condições iniciais definidas.

Uma vez que o ADC tem uma resolução de conversão de 4096 (12 bits), isto quer dizer que o sistema poderá ter uma resolução de ângulo de  $0,00043956^\circ$  (Equação 31). Isto apenas será possível se os picos mínimos do sinal sejam de 0V e os picos máximos sejam de 3,3V, o que é difícil de obter devido às limitações na regulação do circuito LED-fototransistor. Considerando estes valores, ter-se-á uma resolução em valores de tensão de  $0,00080586V$  (Equação 32). Para se obter uma resolução máxima é também necessário ter uma frequência de conversão adequada à frequência do sinal sinusoidal que está diretamente ligada à velocidade de rotação do motor.

$$Res_\theta = \frac{\text{Ciclo elétrico}}{Res_{ADC} - 1} \quad (30)$$

$$Res_V = \frac{\text{Amplitude}}{Res_{conv} - 1} \quad (31)$$

Convencionou-se a velocidade de rotação do motor como  $0,1 \text{ rev/s}$  ( $36^\circ/\text{s}$ ). Logo a frequência sinusoidal será de  $20\text{Hz}$  (Equação 33). Isto pressupõe que o ADC seja capaz de fazer, no mínimo, 4096 conversões num intervalo de  $0,05$  segundos (período do sinal elétrico) para se obter a resolução máxima do sinal (Equação 34). O que quer dizer que a frequência de conversão mínima é de  $81920\text{Hz}$ . Para uma primeira iteração utilizou-se  $100\text{kHz}$  como frequência de conversão, que poderá vir a ser aumentado ou diminuído consoante as restrições que se encontrarem.

$$f = \frac{\omega * N}{360} \quad (32)$$

$$f_{convers\tilde{a}o} > f * (Res_{ADC}) \quad (33)$$

Condições iniciais:

- Número de ciclos por revolução (N) - 200
- Velocidade de rotação do motor ( $\omega$ ) - 36°/s
- Frequência do ciclo elétrico ( $f$ ) - 20Hz
- Amplitude máxima do sinal (*Amplitude*) - 3,3V
- Resolução do ADC ( $Res_{ADC}$ ) - 4096
- Resolução em Volt ( $Res_V$ ) - 0,000806V
- Frequência do ADC - 100000Hz

#### **5.1.1.1. Algoritmo das funções**

De forma a fazerem-se cumprir as condições iniciais foi necessário desenvolver-se funções e algoritmos para a configuração dos módulos do microcontrolador.

##### **5.1.1.1.1. Função fontes de relógio**

Como foi descrito no subcapítulo 3.1.1.2, as diferentes funcionalidades utilizam como fonte de relógio uma das 8 derivações do SYSCLK, denominadas PBCLKx, em que x corresponde a um número de 1 a 8. De forma a habilitar estas funções foi construída a função “PBxCLKsys”. Esta função, utilizando os registos da Tabela A2, ativa as 8 fontes de relógio e aplica o divisor PBDIV a cada uma. Com exceção da fonte de relógio PBCLK3 que lhe foi definida a mesma frequência de relógio de SYSCLK (200MHz), foi definida uma frequência de relógio de metade SYSCLK (100MHz) a todas as fontes de relógio PBCLKx.

Nesta função foi ainda habilitada a possibilidade da utilização de múltiplos vetores de interrupção através do subregisto `INTCONbits.MVEC` (PIC32, n.d.)

Em suma ao chamar-se esta função é atribuída a frequência de 200MHz ao `PBCLK3` e 100MHz aos restantes bem como a habilitação de multivetores de interrupção

#### **5.1.1.1.2. Funções Timer**

Para além das funções de serviço de rotina de interrupção foi criada uma função para a configuração de *timers* chamada “*setTimer*”. Esta função possui 3 argumentos: o *timer* a ser configurado, a frequência que se pretende que o timer funcione e serviço de interrupção (se é pretendido ou não). Apenas é possível configurar um *timer* por cada utilização da função, isto é, se for necessário configurar dois *timers* é necessário utilizar duas vezes a função.

O funcionamento desta função está demonstrado pela Figura 57. Consoante a escolha do *timer* no primeiro argumento será utilizado o registo `TxCON`, correspondente para configuração. Os registos do *timer* a ser configurado são limpos, de seguida é escolhido o préescalar como na Figura 58, caso se esteja a configurar o *Timer1*, e como na Figura 59, caso se esteja a configurar um dos outros *Timers*. A seleção do préescalar e o cálculo do `PRx` ao encontro da frequência definida no 2º argumento e conforme a Equação 15. Caso nos argumentos se tenha escolhido a utilização de interrupções, é atribuído o valor 1 aos registos de habilitação, prioridade e subprioridade e o valor zero ao registo de *flag* de forma que não seja ativada a interrupção logo no início do ciclo (Tabela A5 e Tabela A6). Caso se tenha optado nos argumentos pela não utilização de interrupção, estes registos são todos colocados a zero. Por fim, o registo de que inicia o *timer* é de novo limpo de forma a não iniciar o *timer* de imediato.

Foi ainda construída uma função de interrupção para *timer* utilizando o vetor correspondente (Tabela A5), de forma serem chamadas no fim de cada ciclo, caso se opte por ativar o serviço de rotina de interrupção. Nestas funções o registo de *flag* é limpo de forma a permitir que o processamento saia da interrupção.

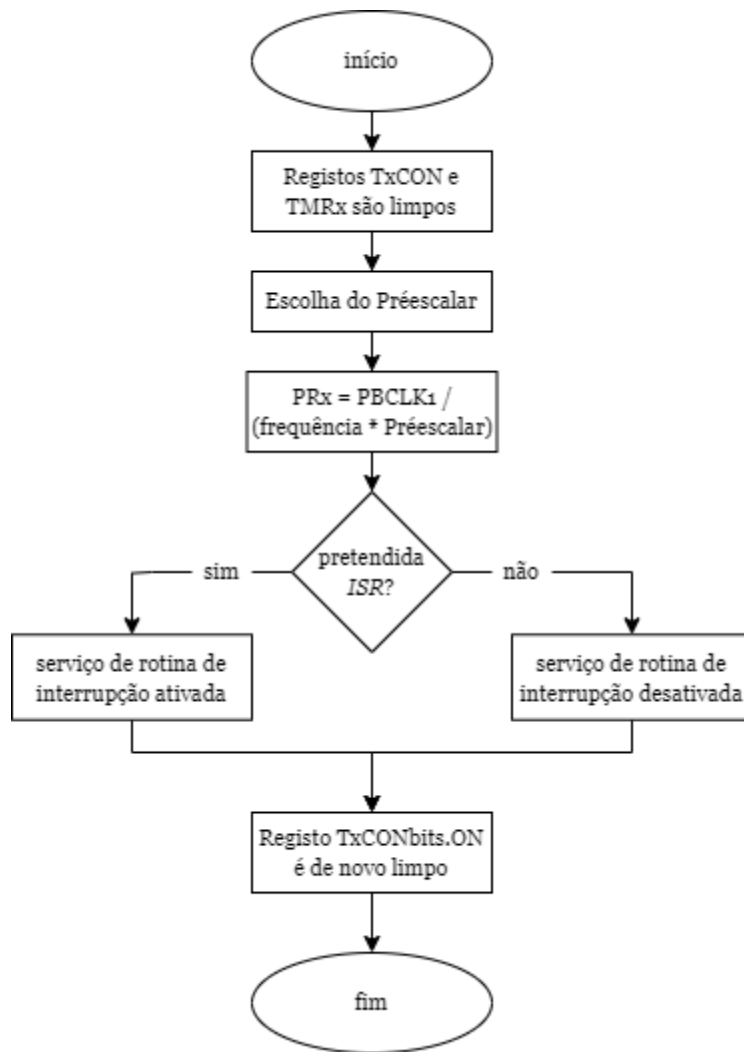


Figura 57 - Fluxograma do funcionamento da função de configuração dos *timers*.

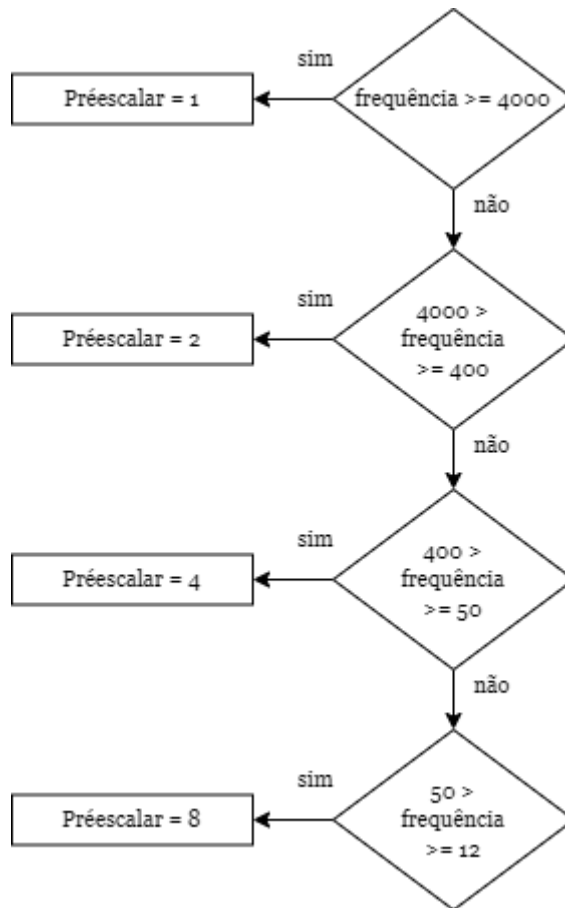


Figura 58 - Escolha do Préescalar para o *Timer1*.

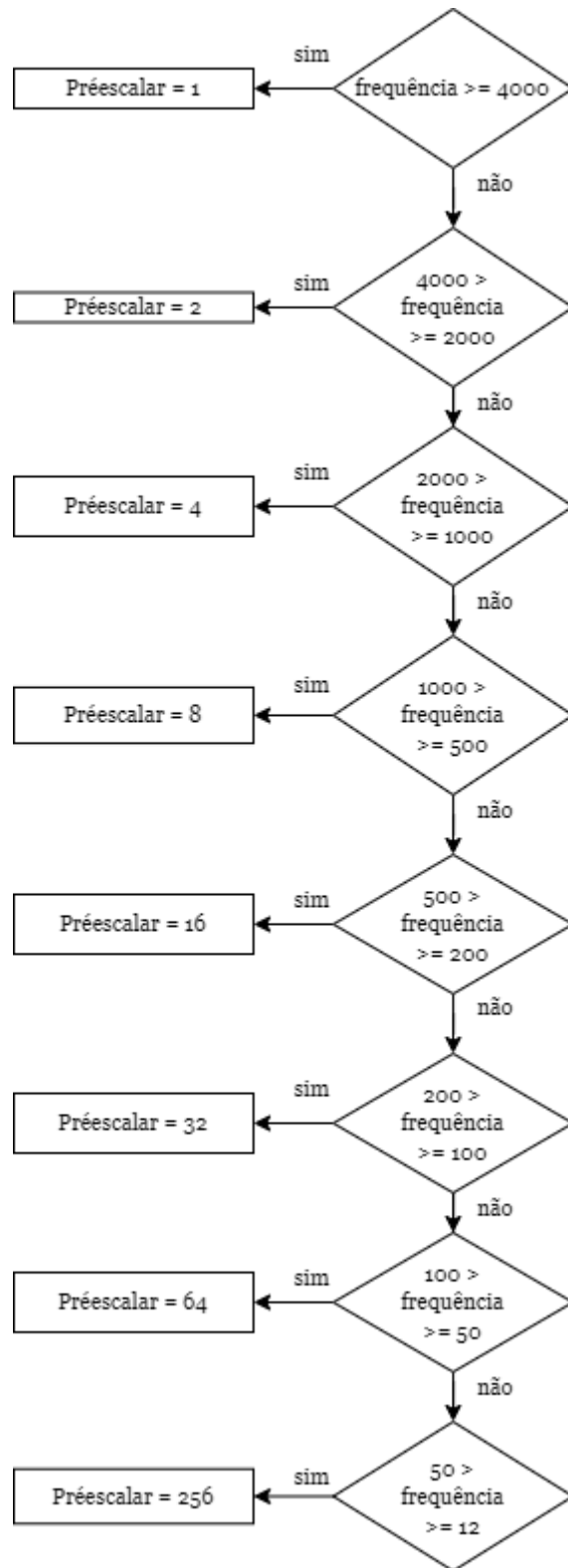


Figura 59 - Escolha do Préescalar do *Timer2* ao *Timer9*.

### 5.1.1.1.3. Funções do comparador de saída

Para além das funções de serviço de rotina de interrupção, foi criada uma função para a configuração de timers chamada “*setOC*”. Embora a função de configuração dos comparadores de saída tenha argumentos que deveriam permitir a configuração dos diferentes módulos para os diferentes módulos, apenas foi criado código para a configuração do comparador de saída 3 no modo de comparação dupla. A função tem ainda argumentos de entrada que permitem escolher o *duty cycle* (em valor percentual) e a utilização ou não de serviço de interrupção. Sendo assim, a Figura 60 apenas contempla o funcionamento da função para a configuração do comparador de saída 3 para o modo de comparação dupla.

Primeiramente, os registos são limpos de forma a evitar que configurações anteriores alterem o comportamento para a configuração agora desejada. Posteriormente é configurado o registo para o efeito em modo comparação dupla e são escolhidos os *timers* 4 e 5 como gatilho (este último será apenas necessário quando necessário um *timer* de 32bits). A configuração do registo OC3RS fará com que o pino de saída passará a nível baixo no valor percentual do *duty cycle* em relação ao registo PR4. Ou seja, se for escolhido um *duty cycle* de 50%, o pino de saída passará a nível baixo a metade do ciclo do *timer*. Uma vez que o registo OC3R é configurado como zero o pino de saída passará a nível alto no início de cada ciclo do *timer*. Uma vez que é necessário o valor de PR4 para configurar OC3RS deverá ser necessário que o *timer* correspondente (neste caso o *timer4*) seja sempre configurado primeiro. Posteriormente é verificado se se optou pela utilização dos serviços de interrupção e os respetivos registos são configurados em conformidade. Este passo funcionará como na função de configuração dos *timers*. A configuração é feita através dos registos da Tabela A7, Tabela A8, Tabela A9 e Tabela A10.

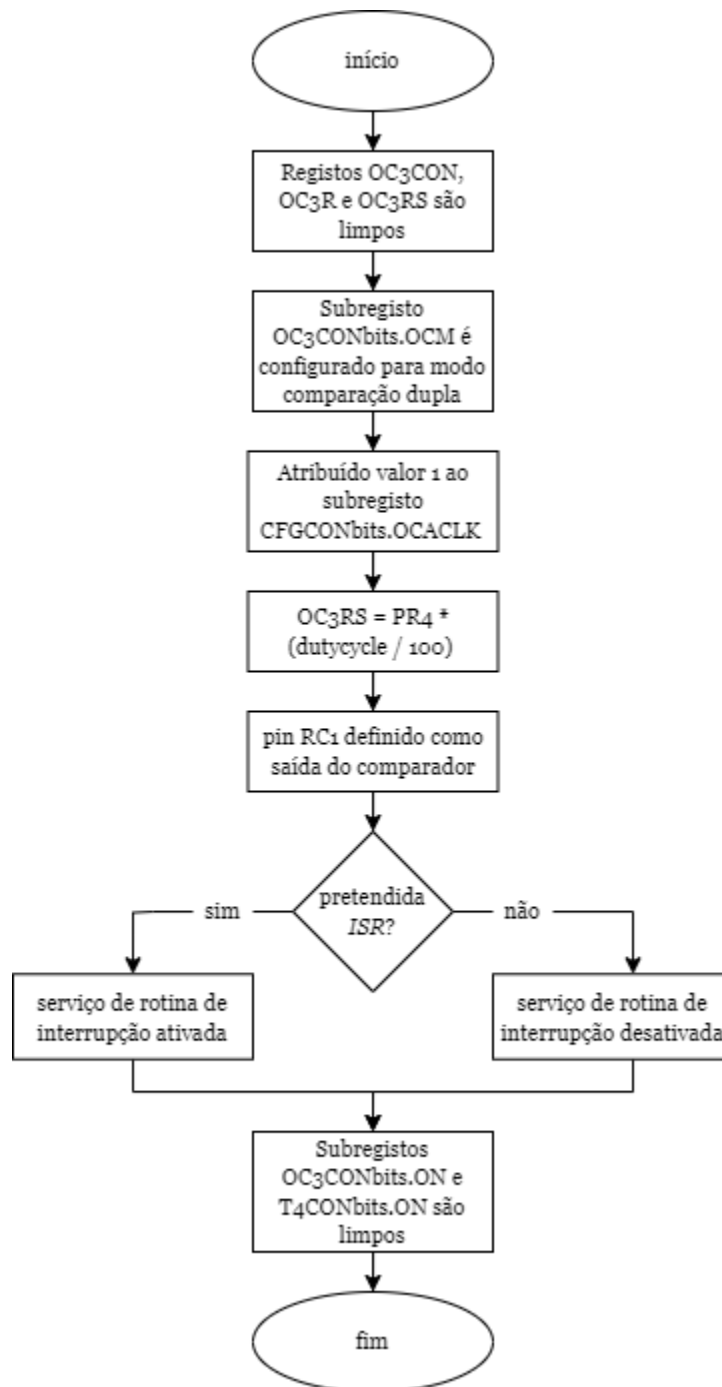


Figura 60 - Fluxograma do funcionamento da função de configuração do comparador de saída 3.

#### 5.1.1.1.4. Funcionamento das funções do conversor analógico digital

Para a configuração do conversor analógico digital foram construídas 3 funções: função de *reset* (“*resetADC*”), função para configuração dos canais (“*setADCchannel*”) e função para configuração do módulo (“*setADCnum*”). A função de módulo.

A função de *reset* servirá apenas para limpar todos os registros ADC e feita a calibração dos registros `ADCxCFG` (página 35).

A função para a configuração dos canais possui dois argumentos para escolher os canais de entrada, ou seja permite configurar dois canais de entrada por utilização da função. Tem ainda um argumento para escolher o gatilho da conversão, um argumento para escolher se a conversão é sensível ao *rising edge* ou nível alto do gatilho e um argumento de forma a habilitar ou não a interrupção. O fluxograma do funcionamento está demonstrado na Figura 61. Primeiramente é configurado o gatilho do primeiro canal selecionado através do registro `ADCTRGx` (Tabela A21). Posteriormente é configurada a conversão para dados com sinal através do registro `ADCIMCONx` (página 33). De seguida é configurado o respetivo pino do canal como entrada analógica através dos registros `ANSELY` e `TRISy` (página 22) e por fim é configurado o serviço de interrupção através dos registros `ADCGIRQEN`, `IPCx`, `IFSx` e `IECx` (páginas 34 e 35). Por fim é repetido o processo para o 2º canal a configurar. Uma vez que não é alterado o registro `ADCIMCONxbits.DIFFx`, esta função não permite o funcionamento em modo diferencial.

A função do módulo embora configure alguns atributos, tem apenas um argumento de entrada que serve para definir que módulo se pretende utilizar. O funcionamento da função está demonstrado na Figura 62. No fundo esta função apenas habilita o módulo, define a tensão de referência como referência interna e define o tempo de amostragem e fonte de relógio através dos registros da Tabela A16, da Tabela A17, da Tabela A18 e do registro `ADCANCON`. A escolha do módulo ADC a utilizar deve estar em concordância com os canais que se configuram na função de configuração de canais (Tabela A20).

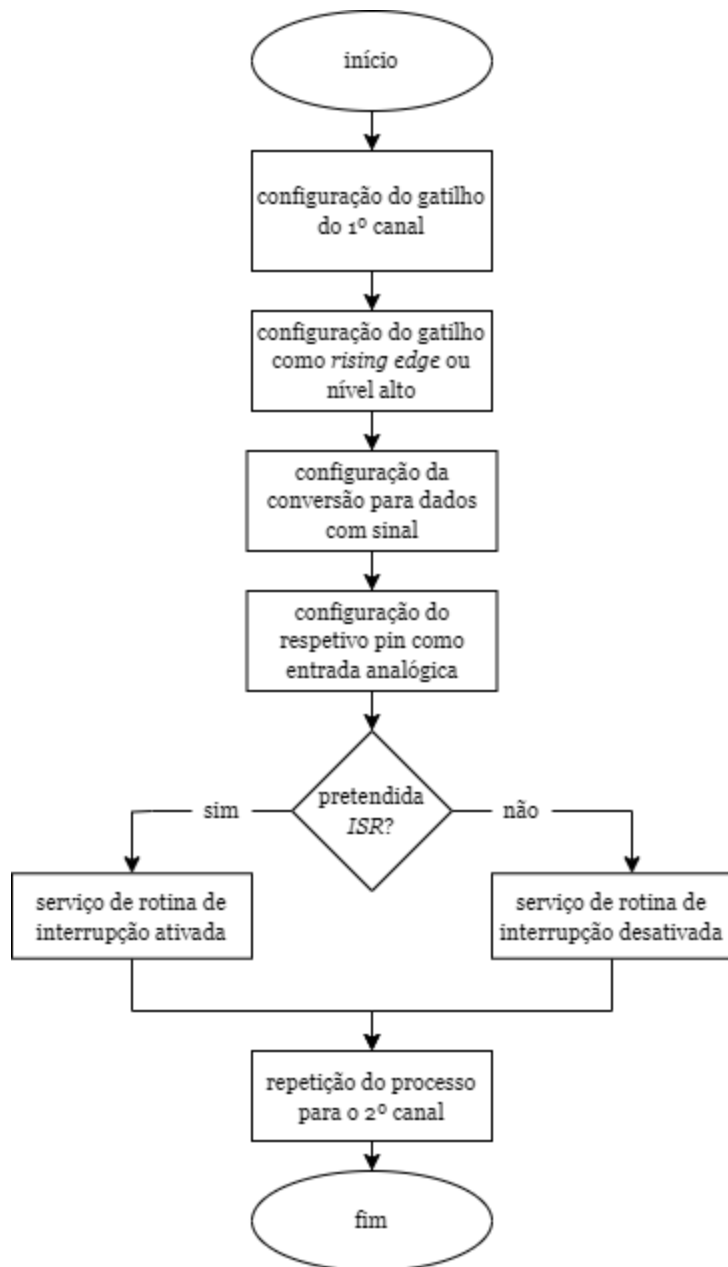


Figura 61 - Fluxograma da função de configuração dos canais do ADC.

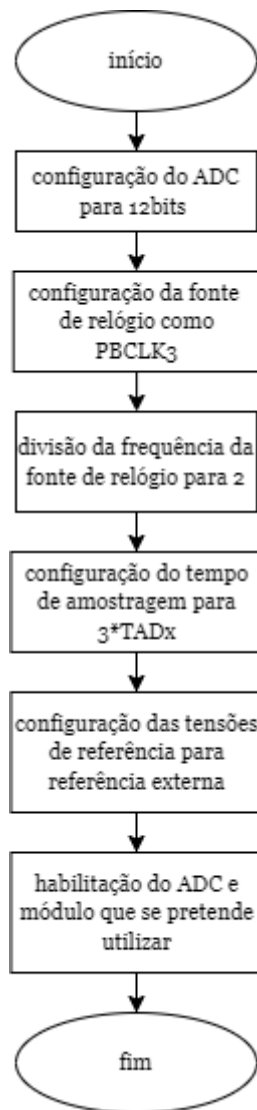


Figura 62 - Fluxograma da função de módulo ADC.

#### 5.1.1.1.5. Funções UART

Foram construídas 2 funções para a UART para além das funções de interrupção: uma função de configuração (“setUART”) e uma para o envio de dados (“senddata”). A função de configuração tem como argumentos: a taxa de transmissão, escolha do módulo UART, escolha da paridade e escolha do número de bits de paragem. O 1º argumento permite escolher uma taxa de transmissão de 9600bps, 57600bps, 115260bps ou 1000000bps. A escolha de paridade é feita através dos subregisto PDSEL e a escolha de bits de paragem é feita através do subregisto STSEL (Tabela A11).

Embora a função tenha sido construída para configurar os diferentes módulos UART a Figura 63, apenas demonstra o funcionamento quando se escolhe configurar o módulo UART2, uma vez que é o módulo utilizado na presente dissertação. A taxa de transmissão é escolhida através do subregistro `BRGH` e do registro `BRGH` de encontro às equações 19 e 20 (página 27). De seguida é definido o pino RF2 como saída da UART e habilitado o modo de saída e desabilitado o modo de entrada UART (Tabela A12). Os pinos RA2 e RA3 são também configurados como saídas e colocados a nível alto de forma a permitir o envio de dados através do *transceiver* MAX485E. Para a UART é apenas habilitada o serviço de rotina de interrupção de erro de forma a saber-se se houve algum erro na comunicação da “mensagem” (Tabela A14). Por fim, o registro `U2MODEbits.ON` é limpo para que a comunicação UART não comece de imediato.

O envio de dados fica a cargo da função “`senddata`”, representada na Figura 64. Uma vez que os dados adquiridos pelo ADC terão 12 bits, é necessário enviar cada dado dividido em dois. Por segurança é verificado se a comunicação anterior está terminada e se o *buffer* está vazio através dos subregistros `UTXBF` e `TRMT` (Tabela A12). Se estas condições se verificarem, o registro `U2TXREG` (registro de envio) é carregado com os 8 bits mais significativos da amostra a enviar e de seguida são novamente verificadas as condições para se enviarem os 8 bits menos significativos. Esta função tem um único argumento que corresponde à amostra a ser enviada pela UART.

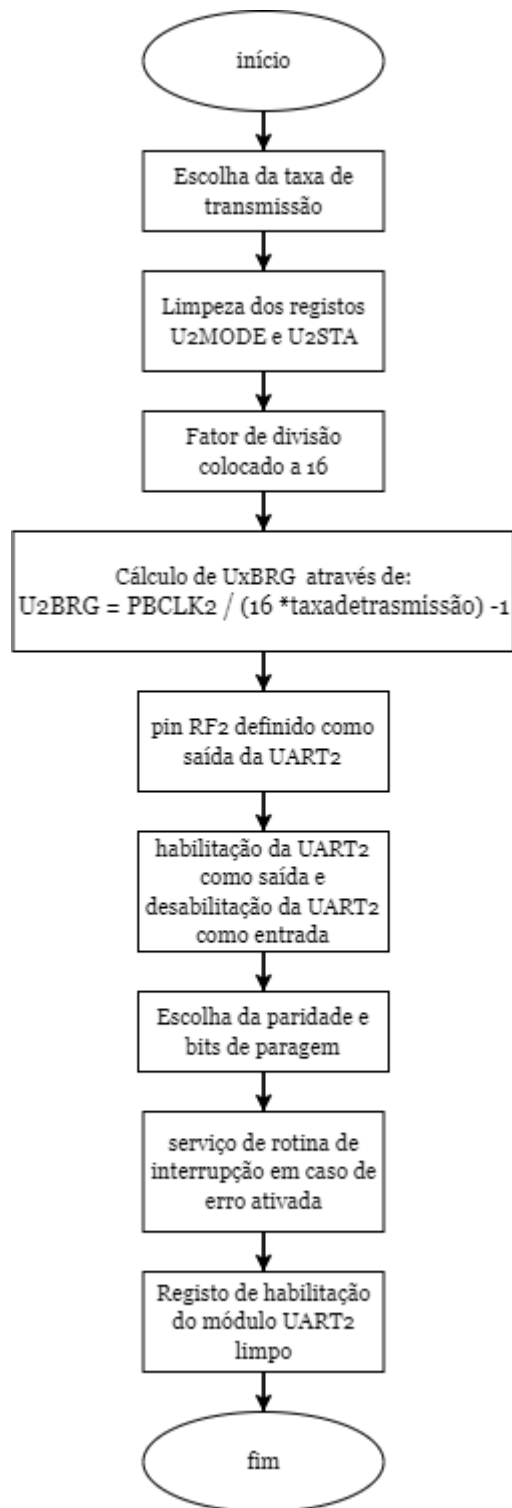


Figura 63 - Fluxograma da função de configuração da UART2.

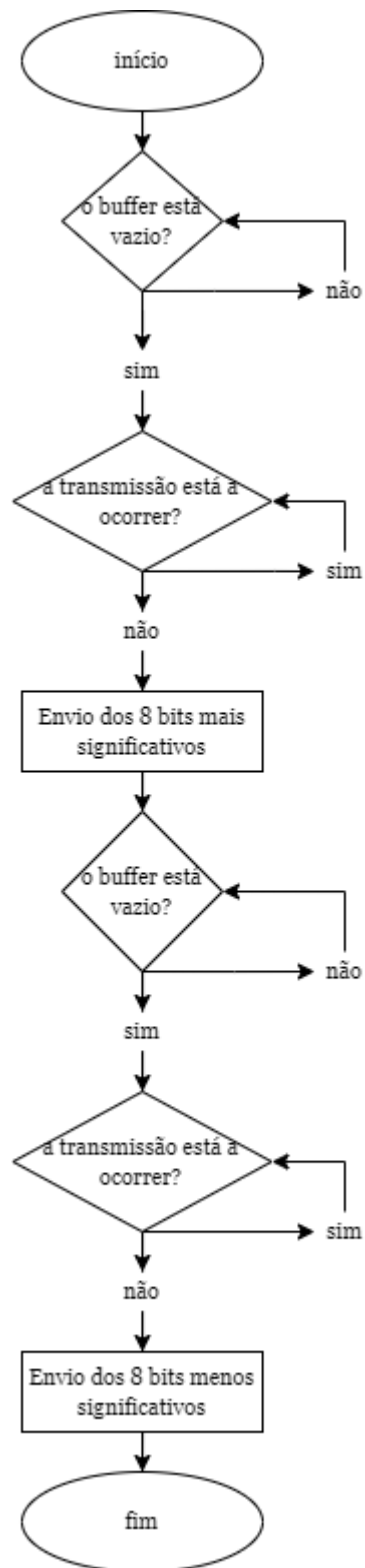


Figura 64 - Fluxograma da função encarregue de enviar dados por comunicação UART.

#### 5.1.1.1.6. Configuração do Driver do motor de passo

A configuração dos pinos que estão conectados ao *driver* foi feita consoante a (Tabela 5). O pino RC1 foi escolhido como pino de passo uma vez que pode ser utilizado pelo comparador de saída 3 e 6 (configurada como saída do comparador 3 na função de configuração dos comparadores de saída). A configuração destes pinos (à exceção do pin RC1, que é configurado na função que configura o comparador de saída), é feita por uma função denominada “*driver\_config*”. Esta função primeiramente configura os pinos RG8, RG7, RG6, RE7 e RG15 como saídas através do registo `TRISxbits.TRISxy` (*x* corresponde à letra da porta e *y* corresponde ao número do pino), posteriormente coloca em nível alto os pinos RG8, RG7 e RG6 (MS1, MS2 e MS3, respetivamente) configurando o micropasso do motor para  $0,05625^\circ$  ( $0,9^\circ/16$ , Tabela 2), e os pinos RE7 e RG15 (DIR e !ENABLE, respetivamente) em nível baixo. O nível baixo no !ENABLE faz com que as saídas para o motor sejam habilitadas. O nível baixo no DIR fará com que o motor tenha um movimento no sentido contrário aos ponteiros do relógio.

#### 5.1.1.1.7. Fluxograma

Na Figura 65 e Figura 66 está representado o funcionamento de forma generalizada do código para o microcontrolador para o 1º teste. Primeiramente são configuradas as saídas para o *driver* e de seguida são configurados os módulos do oscilador, ADC, UART, e *timers* para *triggers*, assim como um *timer* para inicialização e comparador de saída, estas funções são configuradas de forma a ir de encontro às condições iniciais. A ordem das funções utilizada e respetivos argumentos foi a seguinte:

- *driver\_config* – sem argumentos
- *PBxCLKsys* – sem argumentos
- *resetADC* – sem argumentos
- *setADCchannel* – AN9, AN10, *timer3*, conversão sensível ao rising edge, interrupção habilitada
- *setADCnum* – ADC7

- *setUART* – UART2, taxa de transmissão de 1000000bps, sem paridade, com 1 bit de paragem
- *setTimer* – *timer3*, interrupção habilitada, frequência de 100000 Hz (timer do ADC)
- *setTimer* – *timer2*, interrupção habilitada, frequência de 1000 Hz (timer para UART)
- *setTimer* – *timer4*, interrupção não habilitada, frequência de 640Hz (timer do comparador de saída)
- *setTimer* – *timer6*, interrupção habilitada, frequência de 100000Hz (timer de inicialização)
- *setOC* – comparador de saída 3, modo de comparação dupla, interrupção habilitada, *duty cycle* de 50%

A frequência de 640Hz no *timer* do comparador de saída pretende cumprir a condição inicial de velocidade de rotação segundo a Equação 35, uma vez que o passo foi configurado para  $0,05625^\circ$ . No entanto, devido à granularidade dos registos do *timer* a frequência imposta será  $640,0082\text{Hz}$  e consequentemente a velocidade angular do motor  $36,00046^\circ/\text{s}$ .

$$\omega = f_{\text{passo}} * \theta_{\text{passo}} \quad (34)$$

Depois disso é utilizada uma máquina de estados para se fazer todo o processo. Dentro da máquina de estados é ativado o *timer* de inicialização. Após 0,2 segundos é desativado e ativado o comparador de saída e o *timer* de gatilho do mesmo para iniciar o movimento do motor. Após 1500 passos do motor que correspondem a  $84,375^\circ$  mecânicos ( $1500 * 0,9/16$ ) é ativado o ADC e o *timer* de gatilho do mesmo e passa-se ao estado de aquisição de dados. Nota: o motor neste estado mantém-se em movimento e só é parado mais à frente. Neste estado são convertidas 100 mil amostras de cada sinal e assim que se chega à amostra 100mil, o comparador de saída, ADC e *timers* de gatilho são desativados parando o movimento e a conversão. Nota: sempre que é convertida uma amostra os valores de `ADCDATA9` e `ADCDATA10` são colocados em variáveis na função de interrupção que posteriormente são guardados em duas *arrays* de 100000 posições, na máquina de estados. É ativado o *timer* de gatilho da comunicação UART, dando-se início ao envio de dados para o computador através do *transceiver*. Apesar da

comunicação UART não necessitar de *timer* como gatilho, foi utilizada variável como *flag* da UART, isto é, sempre que é colocada a um é feito o envio de uma amostra. A *flag* é colocada a um na função de serviço de interrupção do *timer*. Isto tem como propósito apenas espaçar o envio das amostras. Após serem enviados todos os dados convertidos de cada sinal a comunicação UART é desativada e dá-se o fim da máquina de estados.

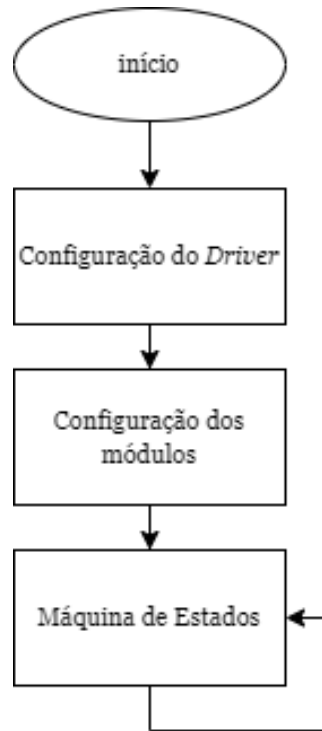


Figura 65 - Fluxograma geral do funcionamento do 1º teste.

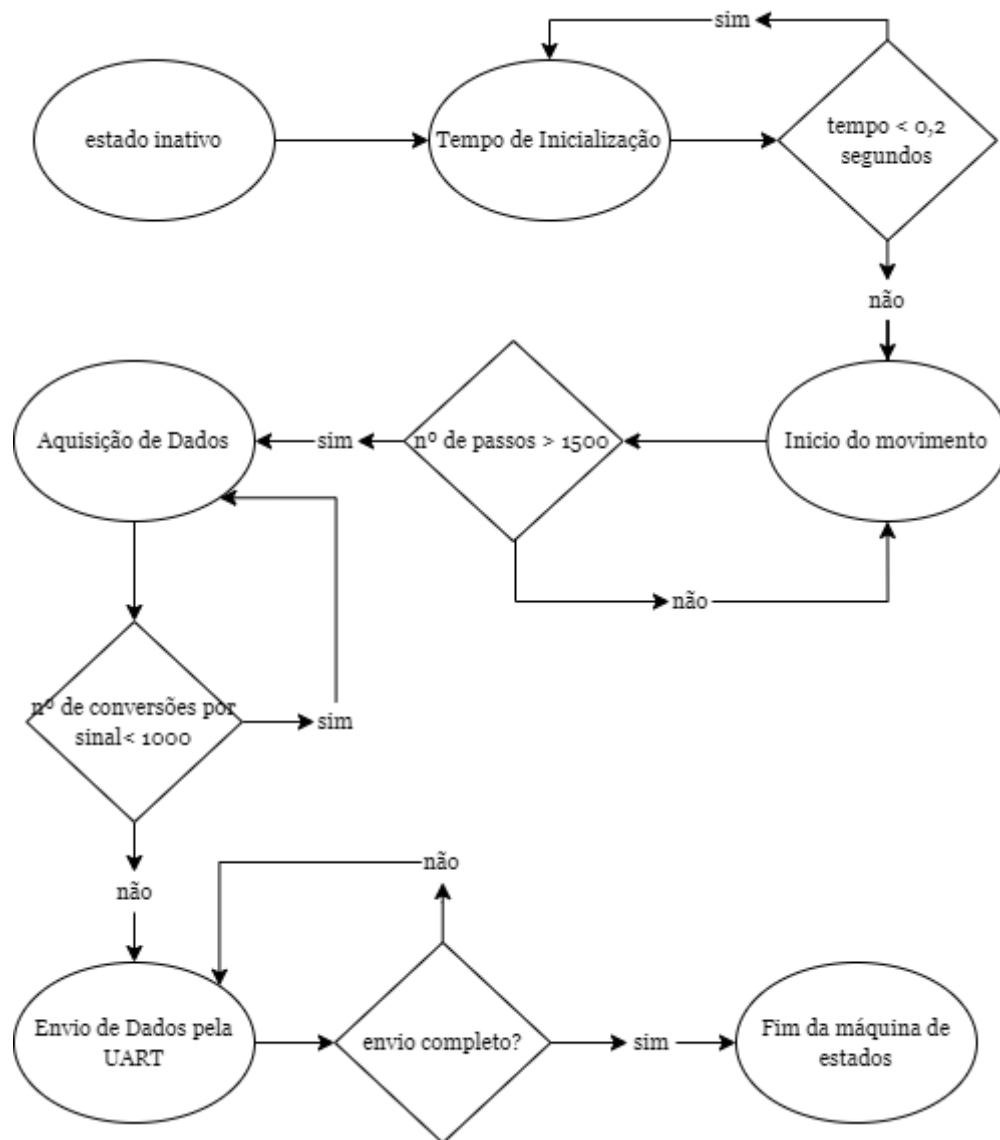


Figura 66 - Fluxograma da máquina de estados do 1º teste.

### 5.1.2. Escolha do Disco

Antes de ser efetuado o primeiro teste com a frequência de amostragem de 100kHz, foi alterada frequência de amostragem para 10kHz para se recolherem dados de todo o perímetro do disco. Uma vez que a máquina de estados foi desenhada para manter o motor em movimento e a conversão do sinal até se obterem 100 mil amostras por sinal, reduzindo a frequência de conversão para este novo valor, obter-se-á amostras de sensivelmente 360° do disco. Esta redução de frequência de conversão deve-se a restrições de memória do microcontrolador, uma vez que este tem de guardar todos os

dados convertidos em duas matrizes (uma para cada sinal) para posteriormente serem enviadas pela UART.

Após a primeira amostra verificou-se que a amplitude e valor médio do sinal recolhido era bastante pouco uniforme o que provavelmente se deveria a irregularidades nos furos do disco (Figura 67). Foram impressos novos discos em diferentes localizações da cama da impressora e com diferentes velocidades de impressão até se chegar ao resultado da Figura 68.

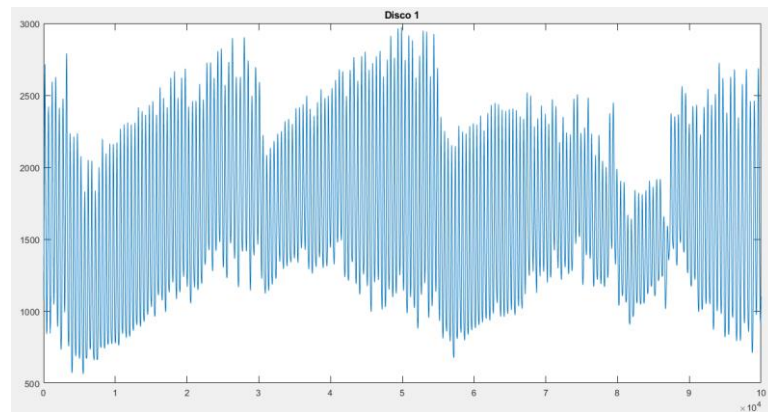


Figura 67 – Sinal sinusoidal de 360° do 1º disco.

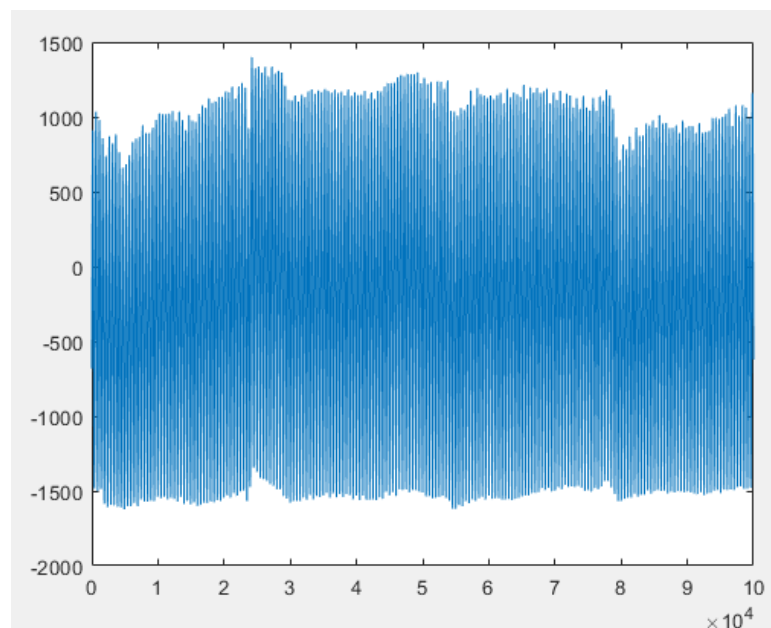


Figura 68 - Sinal sinusoidal de 360° do disco utilizado na dissertação.

### **5.1.3. Resultados do primeiro teste**

Escolhido o disco passou-se à recolha de amostras do 1º teste conforme as condições iniciais definidas. Após ter sido retirada a primeira amostra pôde-se verificar de imediato que a amplitude do sinal ficava bastante aquém dos limites definidos (0V a 3,3V). Sendo assim, verificou-se a necessidade da alteração da tensão de referência para a referência externa, disponível na placa, de 2,5V. Esta primeira amostra apresentou uma amplitude média no valor de 2376 que corresponde a cerca de 1,95V de amplitude. Embora a tensão de referência (teto máximo) seja de 2,5V, permite que a conversão não perca tanta granularidade.

Com a tensão de referência já alterada, foram de novo extraídos dados de forma a ajustar o nível dos potenciômetros para que as duas ondas sinusoidais tivessem sensivelmente a mesma amplitude. Uma vez que se trata de uma afinação manual, a diferença entre as duas ondas pode ser, por vezes, considerável.

Após os ajustes dos potenciômetros, foram retiradas várias amostras como a da Figura 69, para as condições A e B. Para além da diferença de amplitudes que os sinais apresentam, apresentam ainda um valor médio também diferente e ambos descentrados do eixo x, isto é, nenhum dos sinais apresenta um valor médio de zero (necessário para o cálculo do ângulo através do arcotangente). Este último defeito pode ser corrigido através de um filtro digital de passa-alto ou subtraindo a média do sinal ao longo dos 360º (deste método resultarão imperfeições para sinais cujo a média não é regular em toda a extensão, como é o caso).

Já a diferença de amplitude entre os sinais pode dever-se a irregularidades nos furos do disco uma vez que cada par LED-fototransístor se encontra sobre furos diferentes. Se os furos em questão tiverem dimensões diferentes poderão causar amplitudes de onda diferentes. O outro fator que pode causar amplitudes diferentes, já indicado, é um ajuste deficiente dos potenciômetros. Um ajuste digital poderia ter efeitos menos prejudiciais na característica em questão.

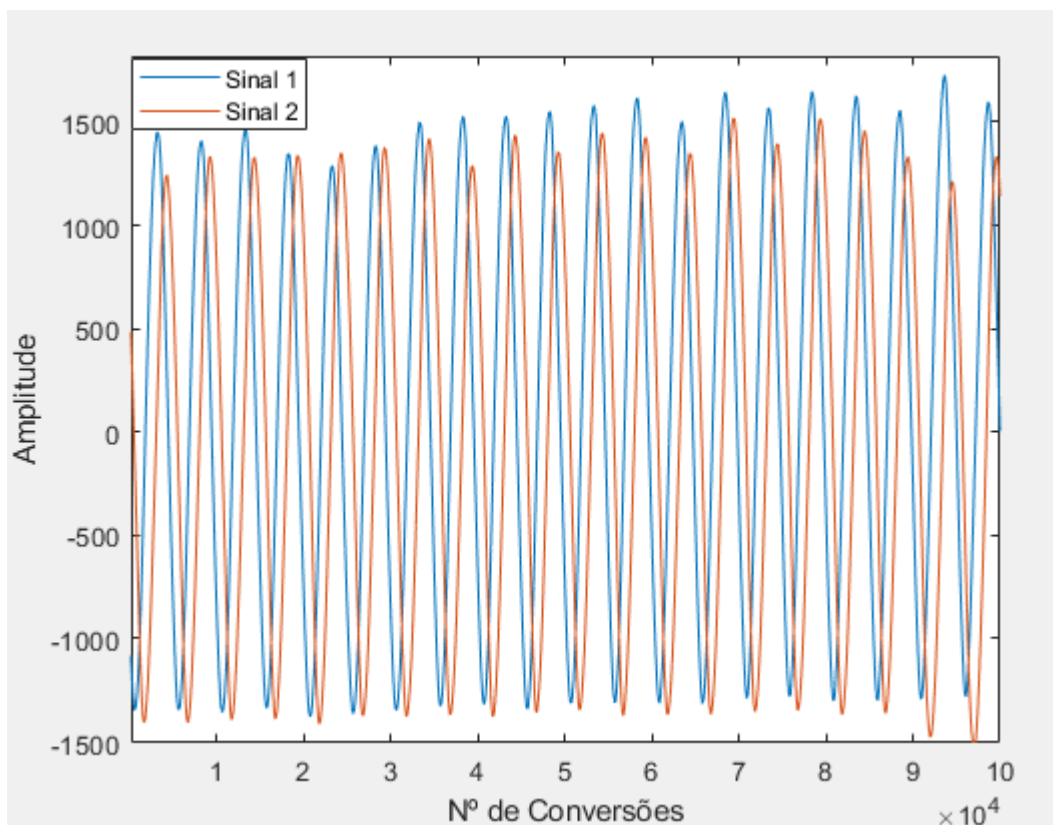


Figura 69 – Amostra de 100mil conversões de cada sinal do 1º teste.

Posteriormente passaram-se as amostras retiradas para as duas condições pela função FFT no Matlab para se analisar a amplitude do espectro das frequências indesejadas. Antes de ser calculada a transformada de Fourier dos sinais, foi aplicada a função BlackmanHarris de forma a reduzir possíveis erros de janelamento.

Em ambos os casos os sinais apresentam harmônicos consideráveis nas frequências de 40Hz, 60Hz, 80Hz e 100Hz (Figura 71). É possível verificar na Figura 70 valores de magnitude bastante consideráveis próximos da frequência zero, o que confirma que os sinais têm um valor médio diferente de zero. Em frequências mais elevadas a magnitude é bastante mais baixa embora se verifiquem alguns picos em algumas frequências (Figura 70). Parte destes picos pode vir a ser eliminado através de filtros digitais.

Na Figura 72 pode-se comparar a diferença de magnitude entre a condição B e A, onde a condição A apresenta, naturalmente, magnitudes mais baixas a altas frequências mas não muito significativas.

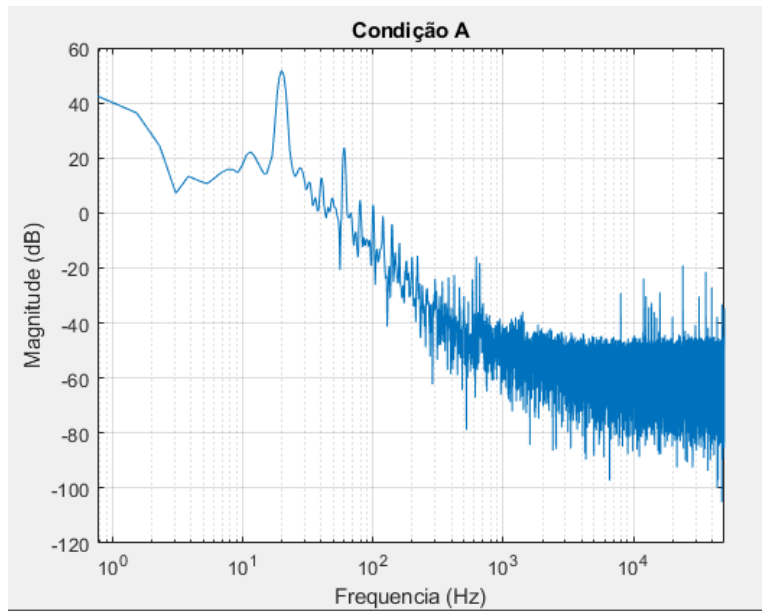


Figura 70 – Gráfico da magnitude de frequências da condição A.

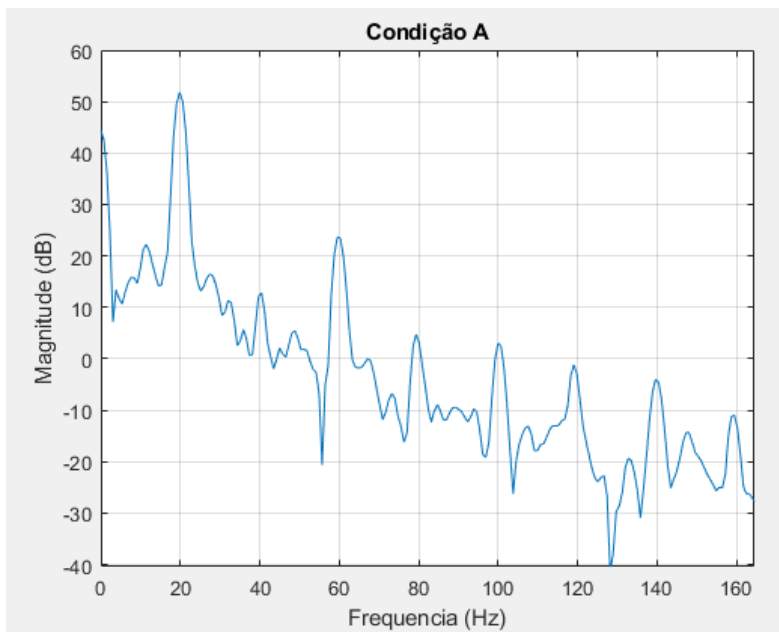


Figura 71 - Gráfico da magnitude de baixas frequências da condição A.

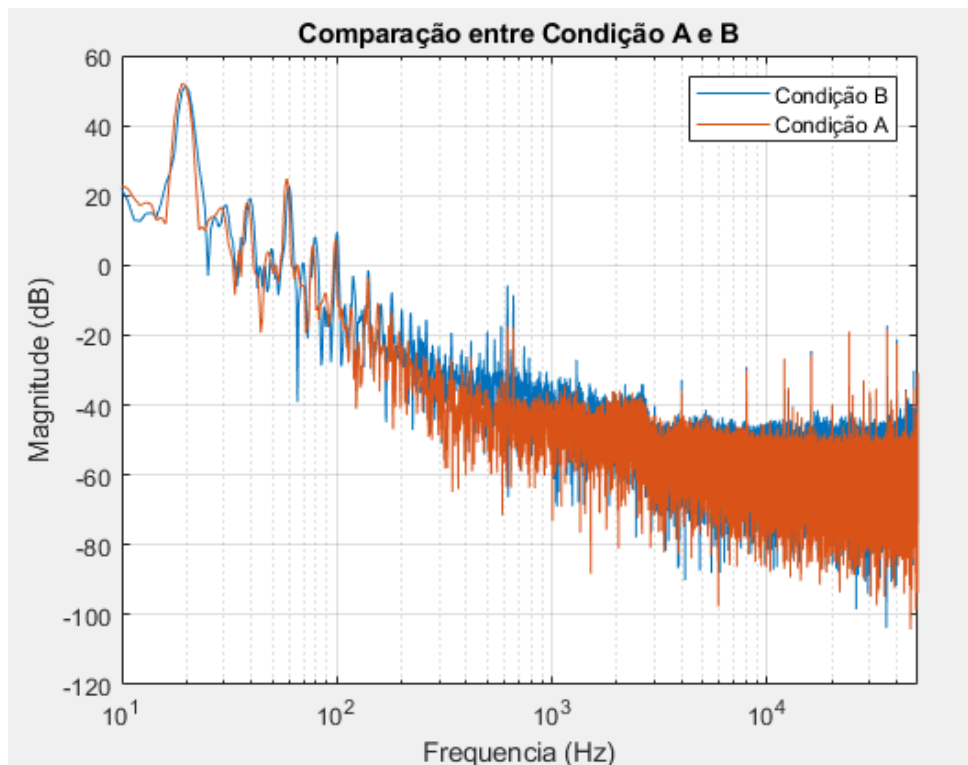


Figura 72 – Comparação da magnitude das frequências entre a condição A e B.

#### 5.1.4. Resolução de problemas do 1º teste

##### 5.1.4.1. Centrar sinais no eixo x

Como foi mencionado anteriormente existem dois métodos que permitem remover a componente DC e centrar os sinais no eixo x: aplicando um filtro digital que remove esta componente ou simplesmente subtraindo a média do sinal.

Para se remover a componente DC de um sinal poderá utilizar-se a Equação 36, uma vez que, para um filtro IIR de passa-alto de 1ª ordem com frequência de corte 1Hz (para se remover a componente DC) os dois coeficientes de b serão iguais em módulo sendo que o 2º será negativo.

Para o 2º coeficiente (o único a entrar na equação de filtragem), quanto mais próximo for o valor deste de 1 menor desfasamento causará ao sinal filtrado em relação ao sinal original. Por outro lado, quanto mais próximo for o valor de 1 maior tempo demorará a estabilizar.

$$y(i) = \text{ganho} * (x(i) - x(i - 1)) + \text{coeficiente} * y(i - 1) \quad (35)$$

Sendo assim foram utilizados como coeficientes os valores: 0,9 ; 0,99; 0,999, 0,9999 e 0,99999. Foi calculado o desfasamento em relação ao sinal original obtendo os indexes dos picos da onda e verificou-se um desfasamento de onda de 89,35°, 82,87°, 38,52°, 4,536° e 0,4320°, respetivamente. Os únicos coeficientes com um desfasamento aceitável são os coeficientes com os valores 0,9999 e 0,99999, no entanto, um filtro com este coeficiente demora alguns ciclos elétricos a estabilizar. Foi calculado o valor médio de cada ciclo ao longo dos 200 ciclos elétricos e só ao fim de 5 ciclos para o coeficiente 0,9999 e ao fim de 60 ciclos, para o coeficiente 0,99999, o valor médio é inferior a 5% do valor médio do sinal original.

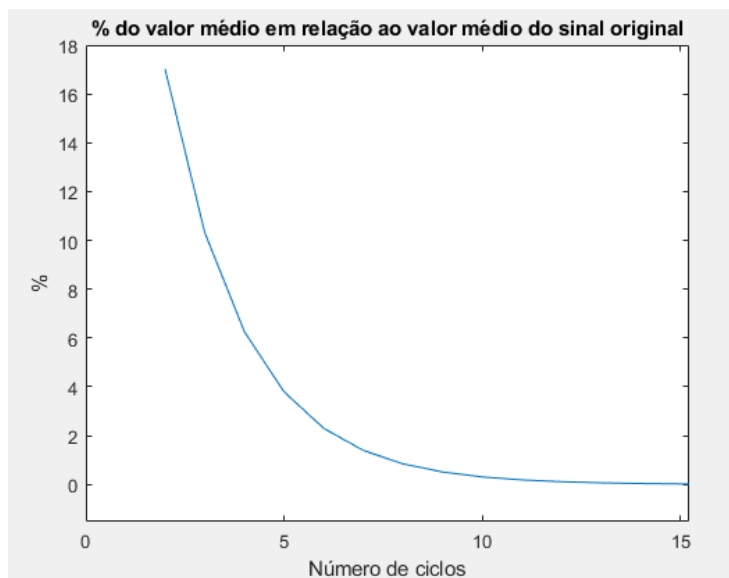


Figura 73 - Número de ciclos que o sinal com o filtro com o coeficiente 0,9999 demora a estabilizar em %.

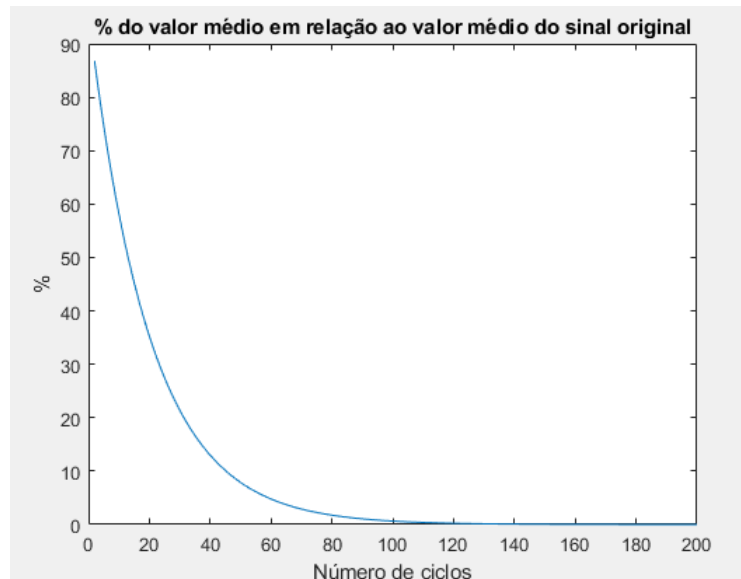


Figura 74 - Número de ciclos que o sinal com o filtro com o coeficiente 0,99999 demora a estabilizar em %.

Posto isto, foram retiradas 3 amostras para cada sinal e para ambas as condições para se calcular a média do sinal ao longo do perímetro do disco. O objetivo pretendido foi comparar o valor médio em cada ciclo elétrico entre a subtração da média do sinal e a utilização do filtro digital com o coeficiente 0,9999. O filtro com o coeficiente 0,99999 foi descartado devido ao número de ciclos que demoraria a estabilizar. O resultado do cálculo das médias foi:

Condição A:

- Sinal 1 : -262.3342
- Sinal 2: 94.0331

Condição B:

- Sinal 1: -233.1042
- Sinal 2: 117.59331

Uma vez que para retirar estas amostras ao longo dos 360° foi necessário reduzir a frequência de conversão para 10kHz. Antes de se aplicar o filtro digital ao sinal, este foi

convertido digitalmente num sinal de 100kHz através da função de Matlab *resample*. Nota: esta função converte digitalmente um sinal retirado numa frequência x para um sinal retirado a uma frequência y e não alterando características como a média e amplitude.

Ao sinal original foi-lhe subtraído o valor médio e o sinal convertido digitalmente num sinal de 100kHz foi filtrado e de seguida foi calculado o valor médio em cada ciclo. Os resultados estão representados nos histogramas da Figura 75 e da Figura 76. Como se poderá ver o sinal filtrado tem valores médios mais próximos do zero.

No entanto, uma vez que o filtro causa um desfasamento que poderá ser considerável e o seu processamento no microcontrolador limitará o número de coeficientes que se utilizarão no filtro digital de passa-baixo, optou-se por utilizar o método de subtração da média.

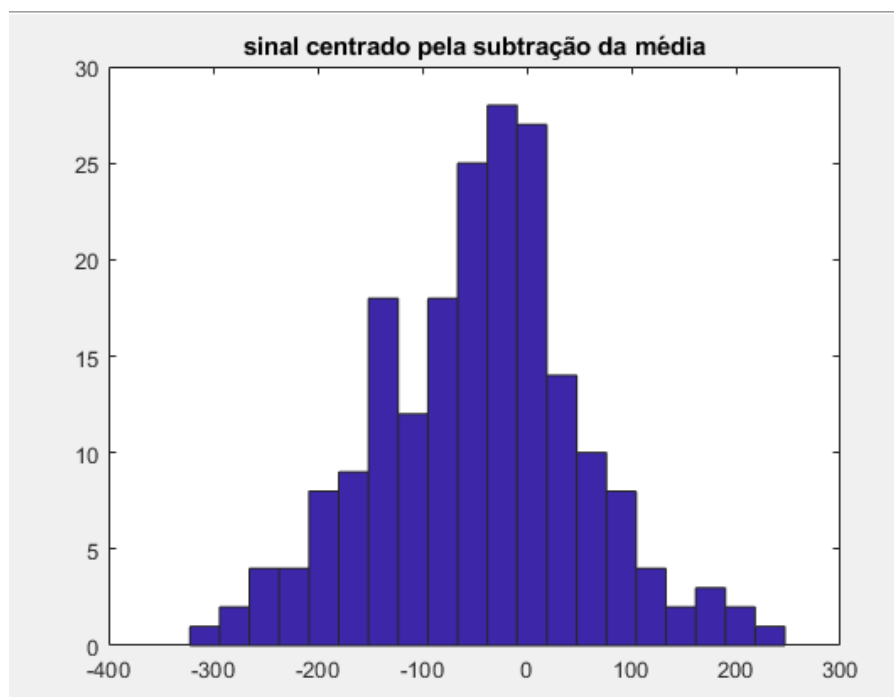


Figura 75 - Histograma de valores médios por ciclo elétrico quando subtraído o valor médio.

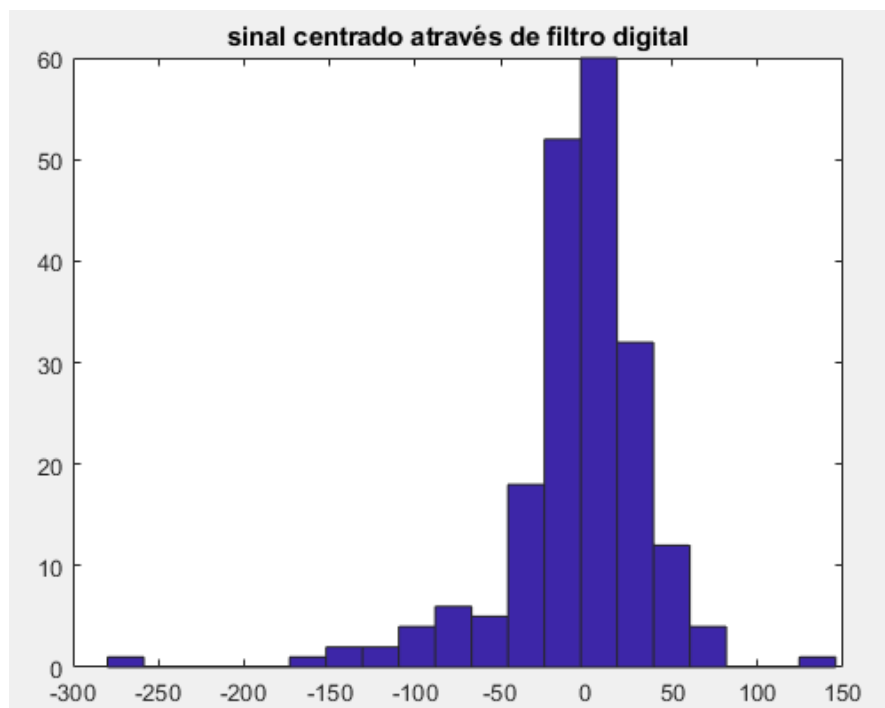


Figura 76 - Histograma de valores médios por ciclo elétrico quando aplicado o filtro digital de passo a alto.

#### 5.1.4.2. Filtro Digital

De forma a reduzir-se o ruído de alta frequência propôs-se a utilização de um filtro digital de passa baixo. Definido anteriormente que o conjunto dos filtros (analógico+digital) não deveria provocar um desfasamento do sinal superior a 1º elétrico. O filtro para a condição A não deve provocar um desfasamento superior a 0,5º elétricos e para a condição B não deve provocar um desfasamento superior a 0,7º elétricos.

Outra limitação do filtro é o tempo de processamento do mesmo. Uma vez que a frequência de conversão é de 100kHz tem-se um intervalo de tempo entre cada conversão de 10 microssegundos. Sendo que o microcontrolador despende de algum tempo na interrupção de conversão e a processar o cálculo do ângulo. Para se aferir quanto tempo se dispunha para os cálculos necessários ao filtro foram colocados pinos a nível alto à entrada da interrupção de conversão e à entrada da função de cálculo do angulo e colocados a nível baixo à saída das mesmas. Estes pinos foram ligados às pontas de prova de um osciloscópio para se fazer a medição de tempos (Figura 77).

De seguida foi criado uma função com um algoritmo de forma a filtrar-se um sinal através de um filtro IIR com N coeficientes. Este algoritmo está representado na Figura

92. Esta função deve ser chamada em cada ciclo de forma a filtrar-se o novo dado convertido.

De forma a saber-se o tempo de processamento desta função colocou-se um pino nível alto à entrada desta função e a nível baixo à saída e conectado esse pino às pontas de prova do osciloscópio. Este teste foi efetuado várias vezes para números de coeficientes diferentes de forma a aferir o número máximo de coeficientes que se poderá utilizar.

Como se poderá observar na Figura 78, na Figura 79 e na Figura 80, o número máximo de coeficientes que se pode utilizar dadas as limitações são 6. No entanto, por segurança optou-se por 5 coeficientes.

Definidas as limitações para o dimensionamento dos filtros para as condições A e B: 5 coeficientes e um desfasamento de 0,5 e 0,7 graus elétricos (para as condições A e B, respetivamente); procedeu-se ao dimensionamento de filtros IIR no Matlab através da ferramenta de dimensão de filtros digitais. Esta ferramenta permite-nos especificar a ordem do filtro, a frequência de corte e o tipo de filtro. Foram então dimensionados filtros IIR de tipo *Butterworth* com várias frequências de corte de forma a aferir o desfasamento que causam a ondas sinusoidais de 20Hz.

Para aferir o desfasamento causado por cada filtro foi construída uma onda sinusoidal X de 20Hz e aplicado cada filtro de forma a criar uma onda Y. A cada uma dessas ondas foi aplicada a transformada de Fourier e a esse resultado a função *angle* de forma a converter a componente de fase do número complexo resultante da transformada de Fourier em ângulo. Por fim, foi subtraído o index 21 do resultante desta função da onda X ao index 21 do resultante da mesma função da onda Y. O index 21 corresponde aos 20Hz. Foram dimensionados vários filtros através da ferramenta do Matlab até se chegar a um desfasamento perto do limite definido. Após várias iterações, a frequência de corte do filtro definida para a condição A foi 5750Hz e para a condição B 4200Hz, sendo desfasamento causado é de 0,5023° elétricos e de 0,7088° elétricos, respetivamente. Os coeficientes de cada filtro são:

Condição A: a = [1 -3.058290834260461 3.593513340835500 -1.910828827957112  
0.386675221346277]; b= [6.918062477627362e-04 0.002767224991051  
0.004150837486576 0.002767224991051 6.918062477627362e-04]

Condição B:  $a = [1 \ -3.311356915590027 \ 4.161402818387994 \ -2.347133579064321 \ 0.500599919379536]$ ;  $b = [2.195151945738591e-04 \ 8.780607782954364e-04 \ 0.001317091167443 \ 8.780607782954364e-04 \ 2.195151945738591e-04]$ ;

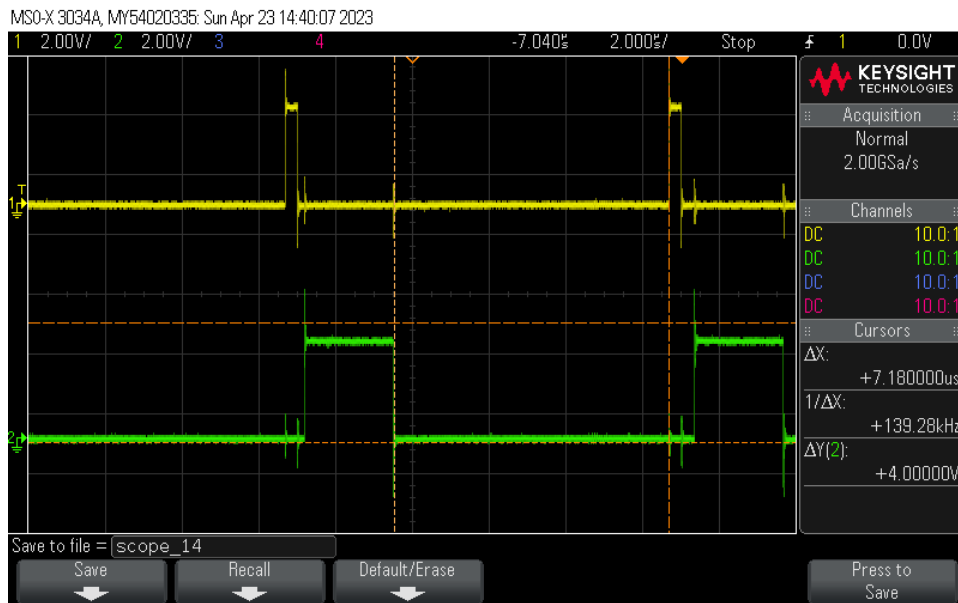


Figura 77 - Intervalo de tempo entre o cálculo do ângulo e o próximo ciclo de conversão. O sinal a amarelo corresponde à interrupção para conversão e o sinal verde corresponde á função de cálculo do ângulo.

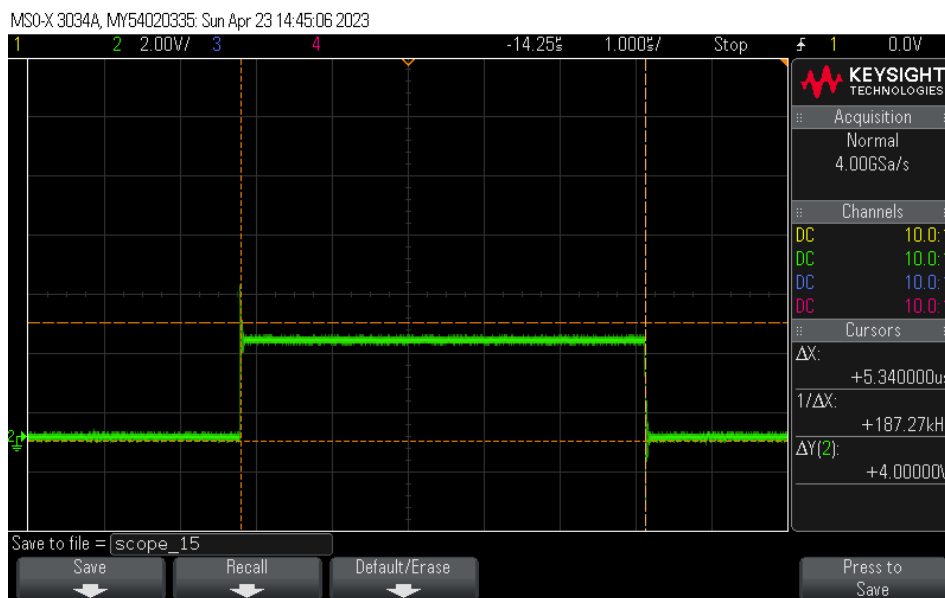


Figura 78 - Tempo de processamento do algoritmo do filtro IIR para 5 coeficientes.

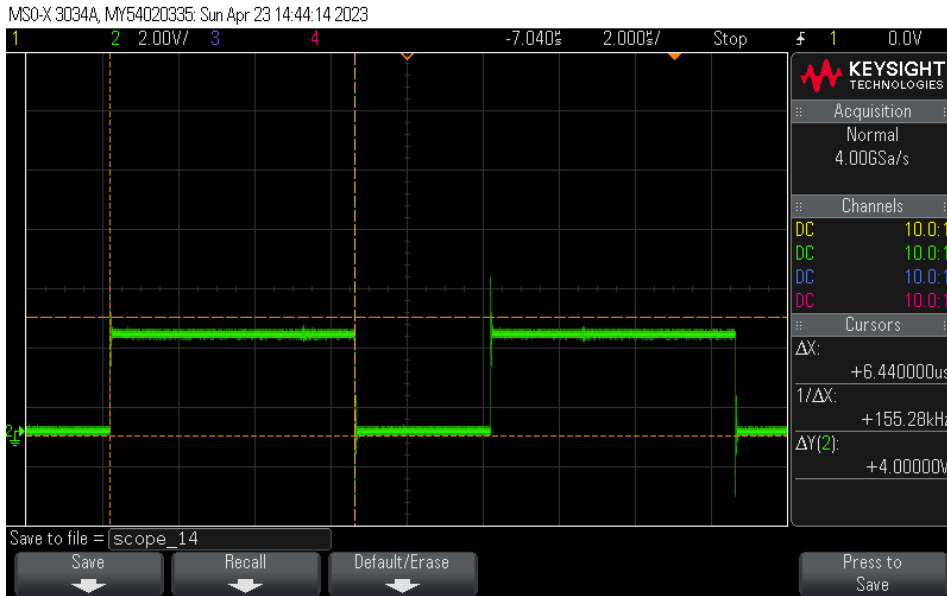


Figura 79 - Tempo de processamento do algoritmo do filtro IIR para 6 coeficientes.

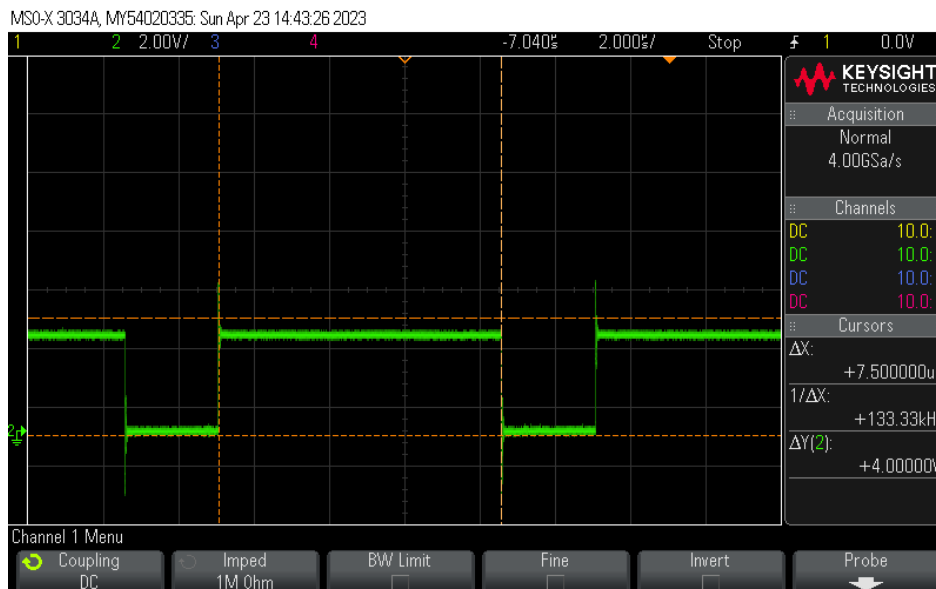


Figura 80 - Tempo de processamento do algoritmo do filtro IIR para 7 coeficientes.

### 5.1.4.3. Simulação dos filtros em Matlab

Escolhidos os filtros, foram aplicados às amostras já retiradas anteriormente, no Matlab e comparadas as magnitudes no domínio da frequência.

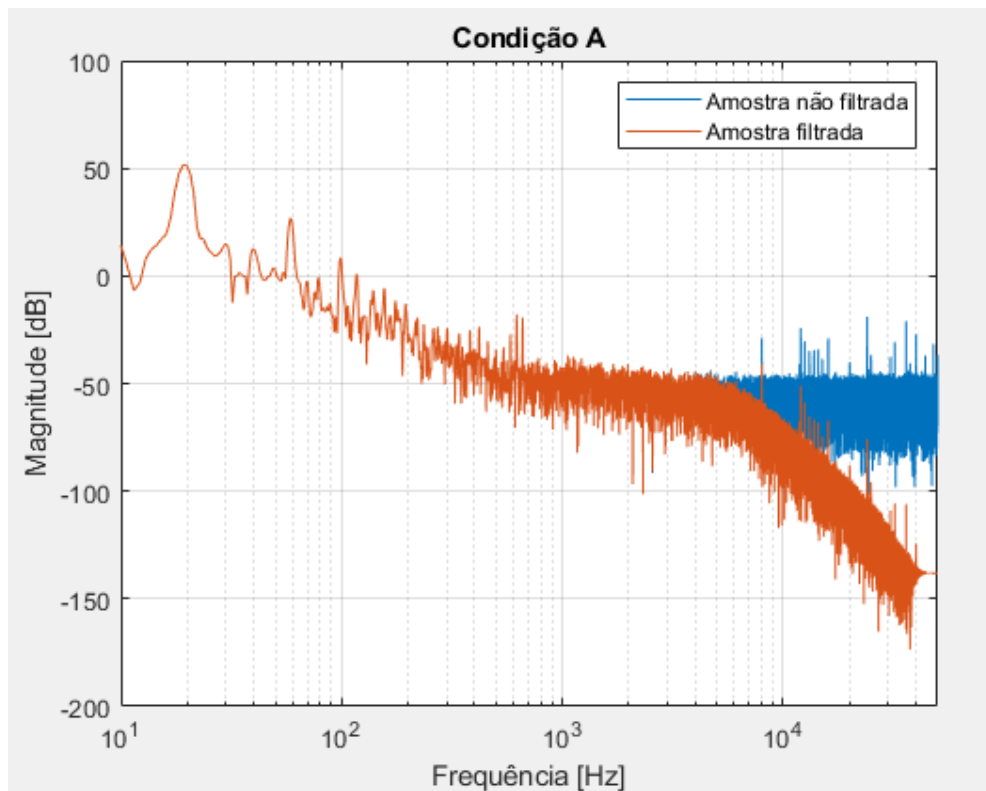


Figura 81 – Comparação da magnitude de frequências entre dados filtrados e não filtrados da condição A.

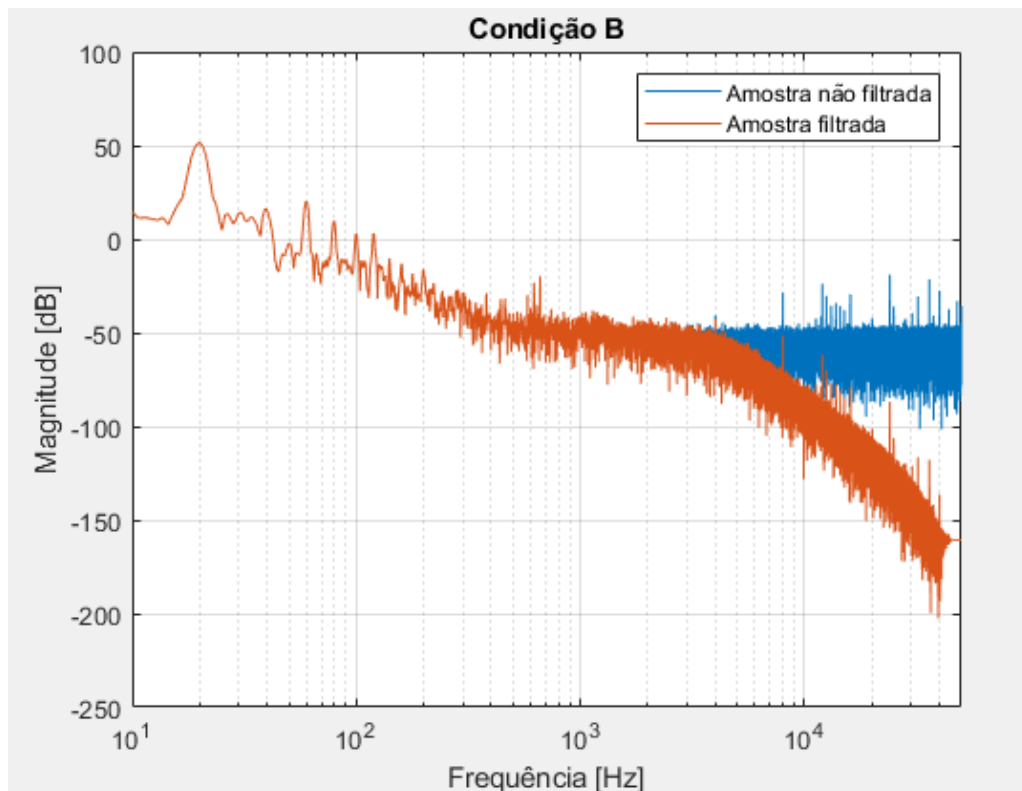


Figura 82 - Comparação da magnitude de frequências entre dados filtrados e não filtrados da condição B.

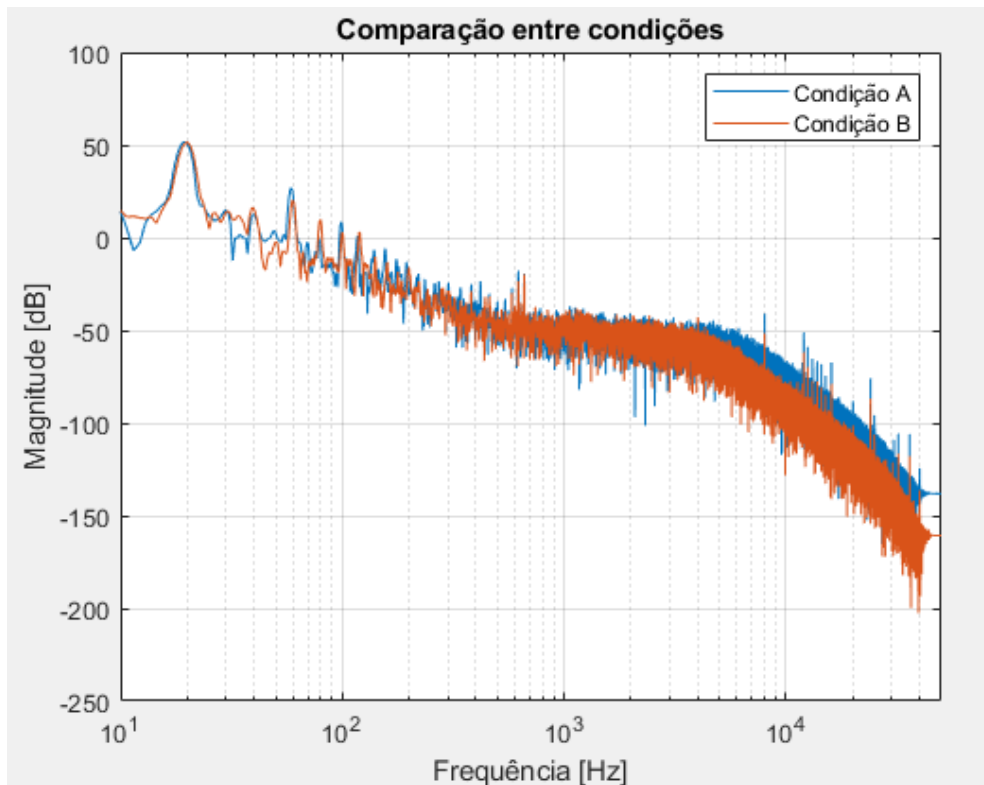


Figura 83 - Comparação da magnitude de frequências dos dados filtrados entre condição A e condição B.

Como esperado, os filtros, tanto para a condição A como para a condição B, apresentam uma redução considerável do ruído de alta frequência. Foi ainda, possível verificar que o conjunto dos filtros (filtro digital + analógico) da condição B apresentam maior redução do ruído que os filtros para a condição A.

Já no domínio do tempo, os sinais filtrados demoram cerca de 25 conversões até estabilizarem, que pode perfeitamente ser contornado com o início do movimento dar-se apenas após algumas conversões.

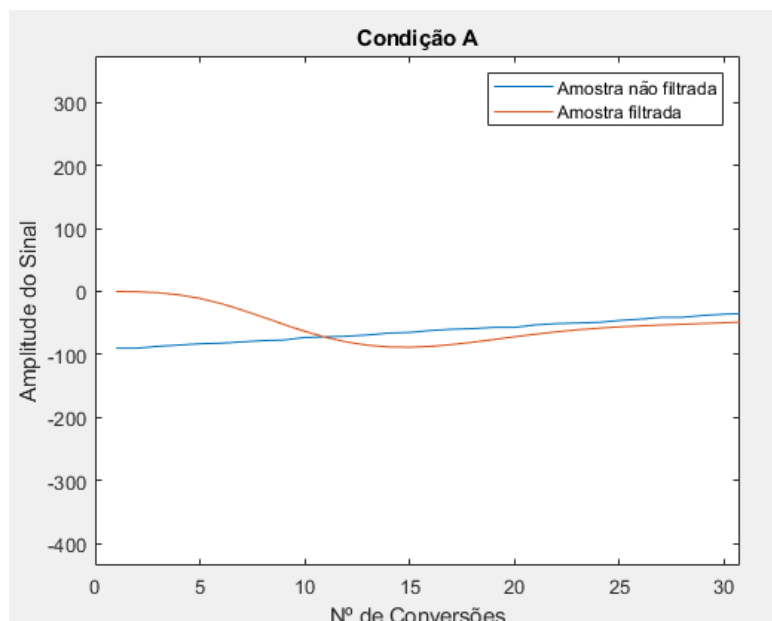


Figura 84 – Comparação das primeiras conversões entre dados filtrados e por filtrar no domínio do tempo.

#### 5.1.4.4. Aplicação do algoritmo do ângulo

Definido o método para centrar os sinais em torno de zero no eixo x, e escolhido o filtro digital de passo a baixo, passaram-se as amostras dos sinais (retiradas no subcapítulo 5.1.3., já filtradas, pelo algoritmo de cálculo do ângulo elétrico no Matlab para se verificar possíveis irregularidades.

Na Figura 85 e na Figura 88 pode-se observar alguma irregularidade na “reta descendente” (para sinais sinusoidais perfeitos em quadratura obter-se-ia uma reta perfeita). Enquanto que na Figura 85 a irregularidade provavelmente se deverá à forma de losango da representação cartesiana dos sinais sinusoidais (Figura 86), na Figura 88 as irregularidades deve-se ao descentramento dos sinais do eixo como se pode observar na Figura 89 e Figura 90.

Estes tipos de irregularidades verificaram-se tanto para a condição A como para a condição B.

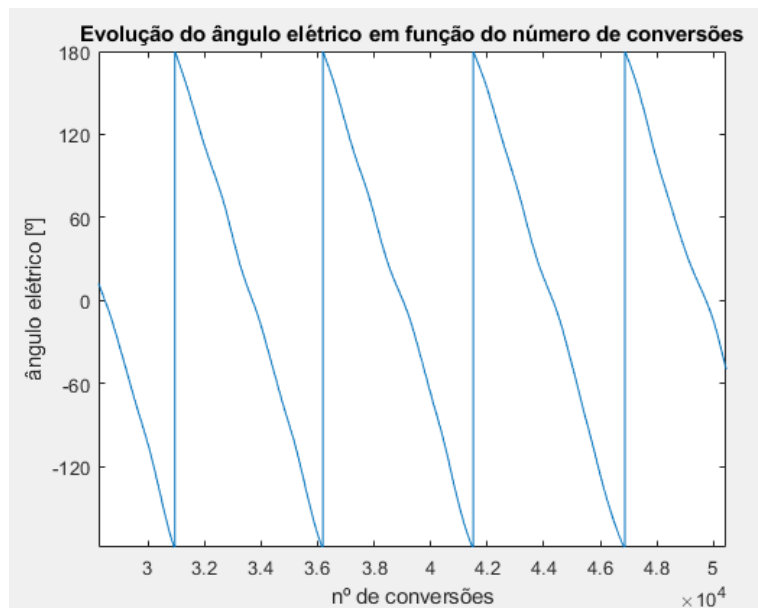


Figura 85 – Gráfico do ângulo elétrico.

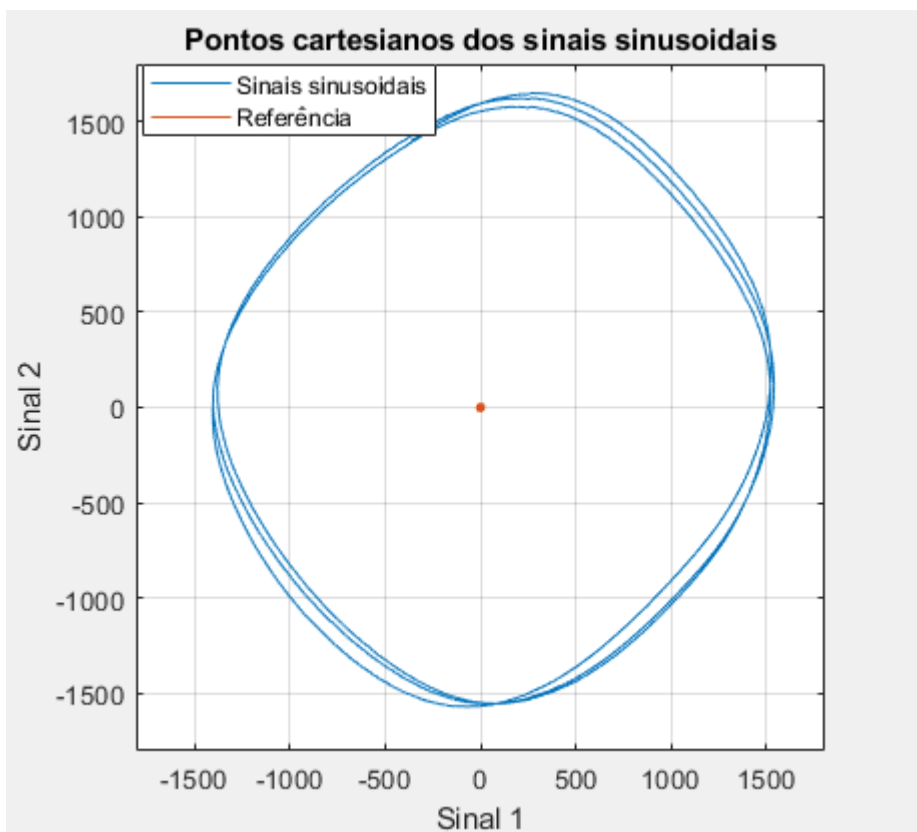


Figura 86 - Representação cartesiana de 3 ciclos elétricos dos sinais sinusoidais da Figura 85.

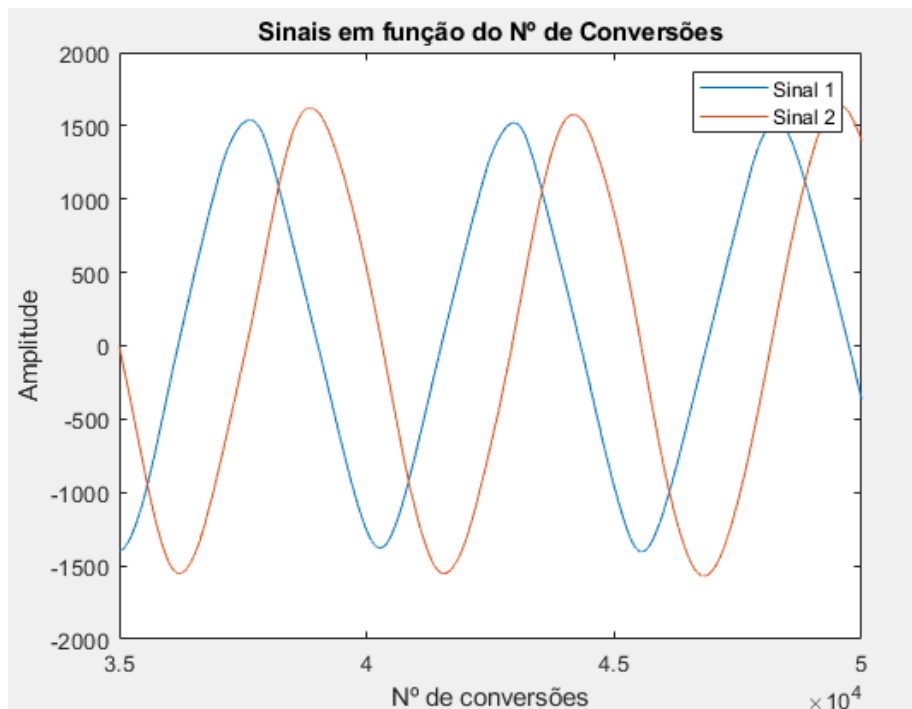


Figura 87 - Representação dos sinais da Figura 85.

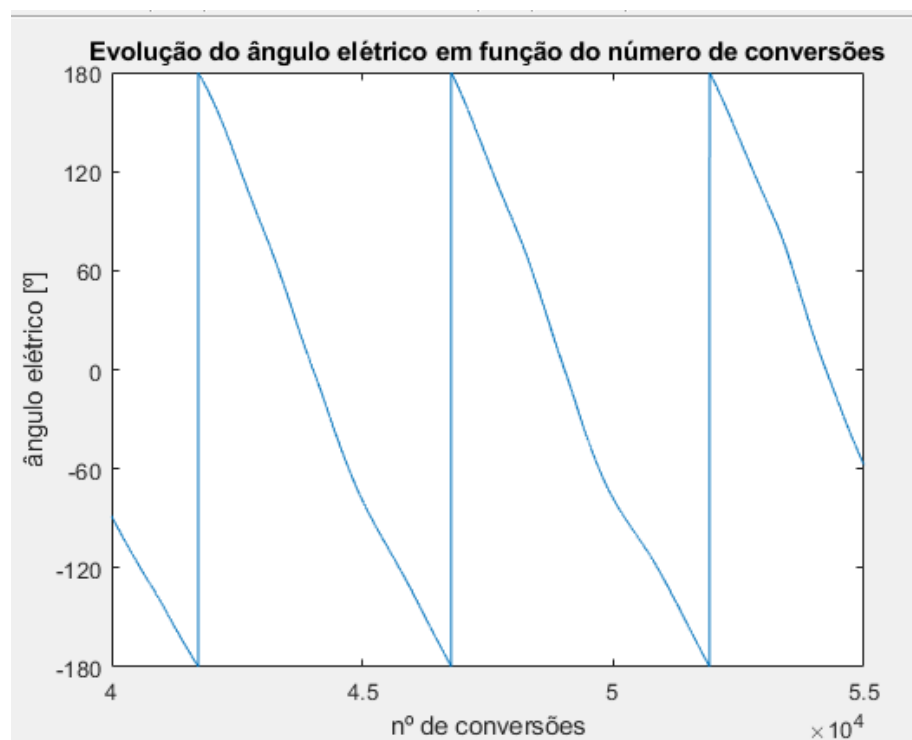


Figura 88 – Gráfico do ângulo elétrico.

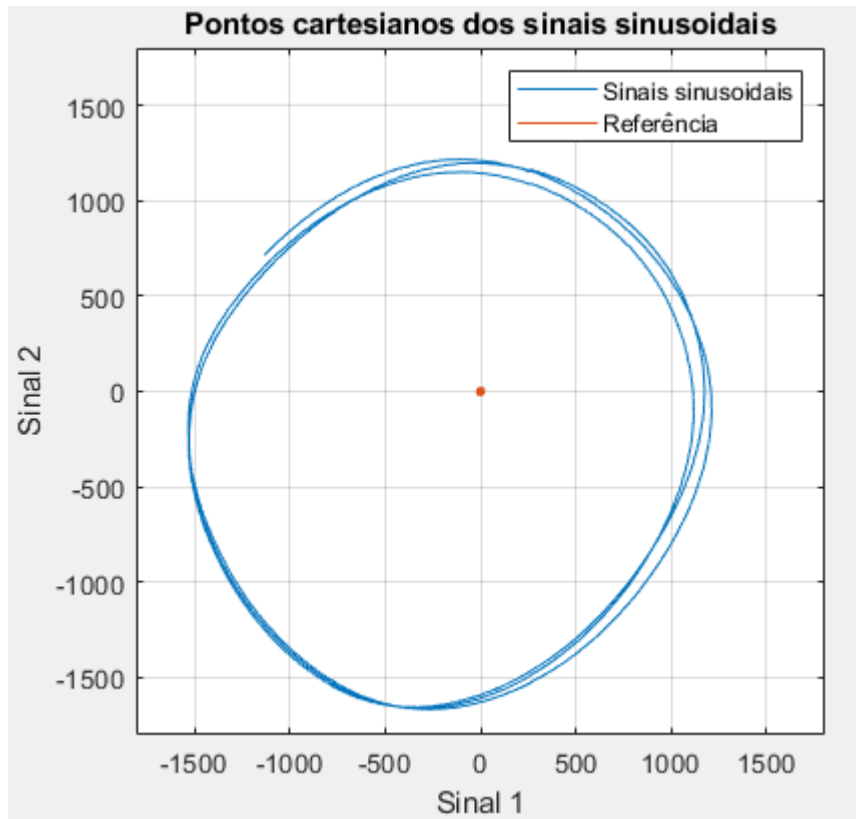


Figura 89 - Representação cartesiana do sinal da Figura 88.

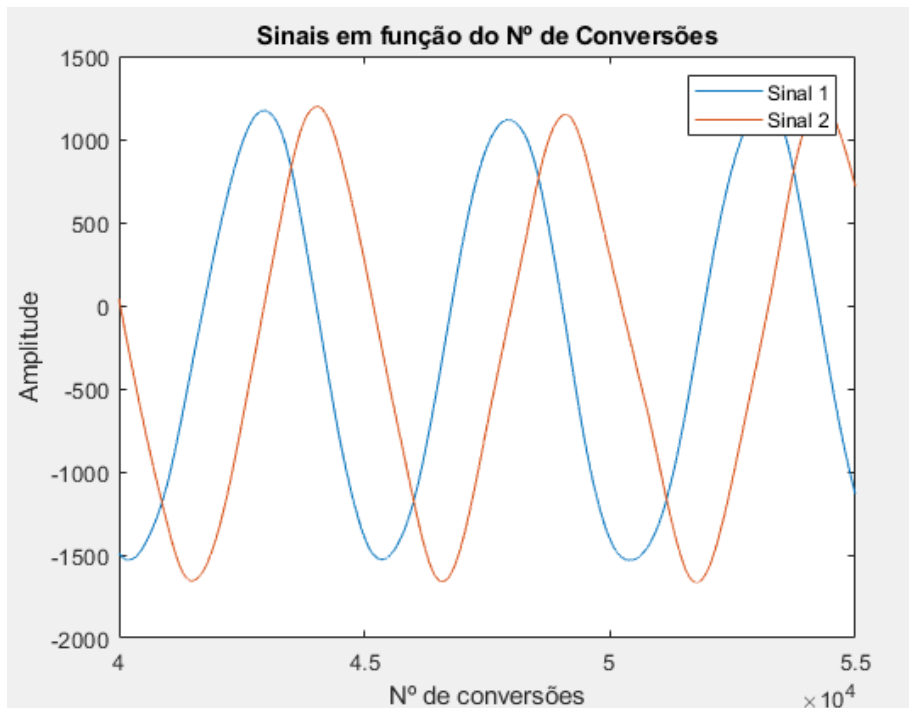


Figura 90 - Representação dos sinais da Figura 88.

### **5.1.5. Recapitulação dos resultados do primeiro teste**

O algoritmo criado para o microcontrolador aparenta estar a funcionar como pretendido. No entanto, poderão observar-se alguns problemas relativos aos sinais convertidos. Estes problemas estão relacionados com o não aproveitamento de toda a resolução do ADC, a diferença de amplitudes dos dois sinais e diferença de amplitude do mesmo sinal ao longo dos 360°.

Em relação ao último ponto, o conversor analógico digital tem uma resolução de 12 bits o que significa que a amplitude de valores convertidos será de 4096. Uma vez que foi feita a conversão para valores com sinal o valor mínimo possível convertido seria -2048 e o valor máximo 2047. No entanto, nas últimas amostras 360 o valor mínimo e máximo para o sinal 1 foi -1667 e 1389, respetivamente e para o sinal 2 -1496 e 1978, respetivamente. Sendo que estes valores mínimos e máximos não correspondem ao mesmo ciclo, a amplitude máxima verificada em cada sinal foi de 2926 para o sinal 1 e 3229 para o sinal 2. Sendo a amplitude média por ciclo de 2629 para o sinal 1 e 2927 para o sinal 2. Isto deve-se à corrente que alimenta o LED uma vez que não permite que o fototransístor devolva o valor máximo a ser convertido e às tensões de referência que não estão ajustadas às amplitudes das tensões a serem convertidas.

Nos dados apresentados, é logo verificável a diferença de amplitudes entre sinal. Para se comprovar, foi calculada a amplitude de cada sinal em cada ciclo e de seguida feita a relação entre as amplitudes por ciclo. A amplitude do sinal 2 é em média 1,113 vezes maior que a amplitude do sinal 1. Esta deficiência deve-se a dois fatores:

1. Regulação deficiente dos potenciômetros, uma vez que é uma regulação manual poderá ter sempre erros consideráveis.
2. Irregularidade dos furos do disco. Dado que os pares LED-fototransístor se encontram sobre furos diferentes encontrar-se-ão, inevitavelmente, sobre dois furos em que um será ligeiramente maior e por isso permitirá maior passagem de radiação que o outro.

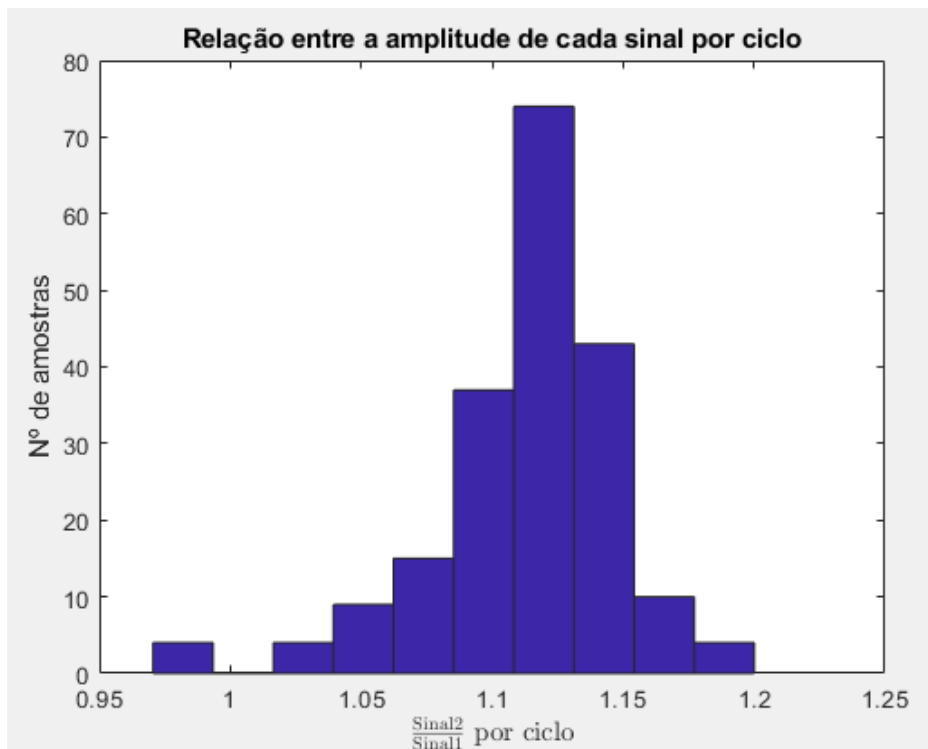


Figura 91 - Histograma da relação entre amplitudes de cada sinal por ciclo.

Foi logo visível na Figura 68 que o sinal não tem uma amplitude regular ao longo dos 360°. Estas irregularidades devem-se a irregularidades nos furos do disco. Esta deficiência não só causará problemas na medição do ângulo devido à diferença de amplitudes das duas leituras. Amplitudes variáveis, e não simétricas em relação ao zero, ao longo dos 360° provocarão um valor médio diferente em cada ciclo o que fará com que introduzirá erros no cálculo do ângulo.

A análise no domínio de frequência, como já referido, permitiu observar harmónicos de magnitude considerável a baixas frequências que poderão também introduzir erros no cálculo do ângulo. Os picos de ruído a altas frequências poderão ser amenizados pelo filtro digital, como já mencionado.

## 5.2. Segundo Teste

Para o segundo teste pretende-se aplicar o filtro digital e o cálculo do ângulo elétrico e mecânico ao algoritmo do microcontrolador. Desse modo, é necessário a criação de novas funções para o efeito.

### 5.2.1. Funções Filtro

De modo a filtrarem-se os dados convertidos a 100kHz através de um filtro IIR, foram desenhadas duas funções. A primeira apenas de inicialização, denominada “*IIRStructInit*”, tem como argumento de entrada um ponteiro para uma estrutura que contém dois *buffers* para entrada e dois *buffers* para saída (para cada sinal) e outras duas matrizes que servirão para os coeficientes *b* e *a* do filtro. Os *buffers* de entrada e saída são também matrizes do mesmo tamanho das matrizes dos coeficientes. Esta primeira função limpa os *buffers* e carrega as matrizes dos coeficientes com os coeficientes pretendidos.

A função principal denominou-se “*Filter\_IIR*” e tem como argumentos de entrada duas variáveis de tipo inteiro de 16 bits, para os valores dos dois sinais acabados de converter, e um ponteiro para uma estrutura do mesmo tipo que a função anterior. Os valores a serem filtrados são passados a uma variável do tipo *double* e, de seguida, são atualizados os indexes dos *buffers*. Isto é, os valores das posições dos *buffers* são movidos para uma posição à direita (o valor que estava no index zero passa para o index 1, e por aí adiante). De seguida, os *buffers* de entrada são carregados no index zero com o valor retirado dos sinais, para depois ser feita a filtragem do sinal através de um *loop* de forma a cumprir a Equação 37. Os dados resultantes do filtro irão encontrar-se no index zero dos *buffers* de saída (*buffer\_y1* e *buffer\_y2*).

$$y(n) = \sum_{k=0}^N ( b(k) * x(n - k) ) - \sum_{k=1}^N ( a(k) * y(n - k) ) \quad (36)$$

Esta função foi desenhada de forma a ser chamada sempre que são convertidos os dados dos dois sinais.

A medição de tempo do processamento para vários coeficientes no capítulo 5.1.4.2, já tem em conta este algoritmo.

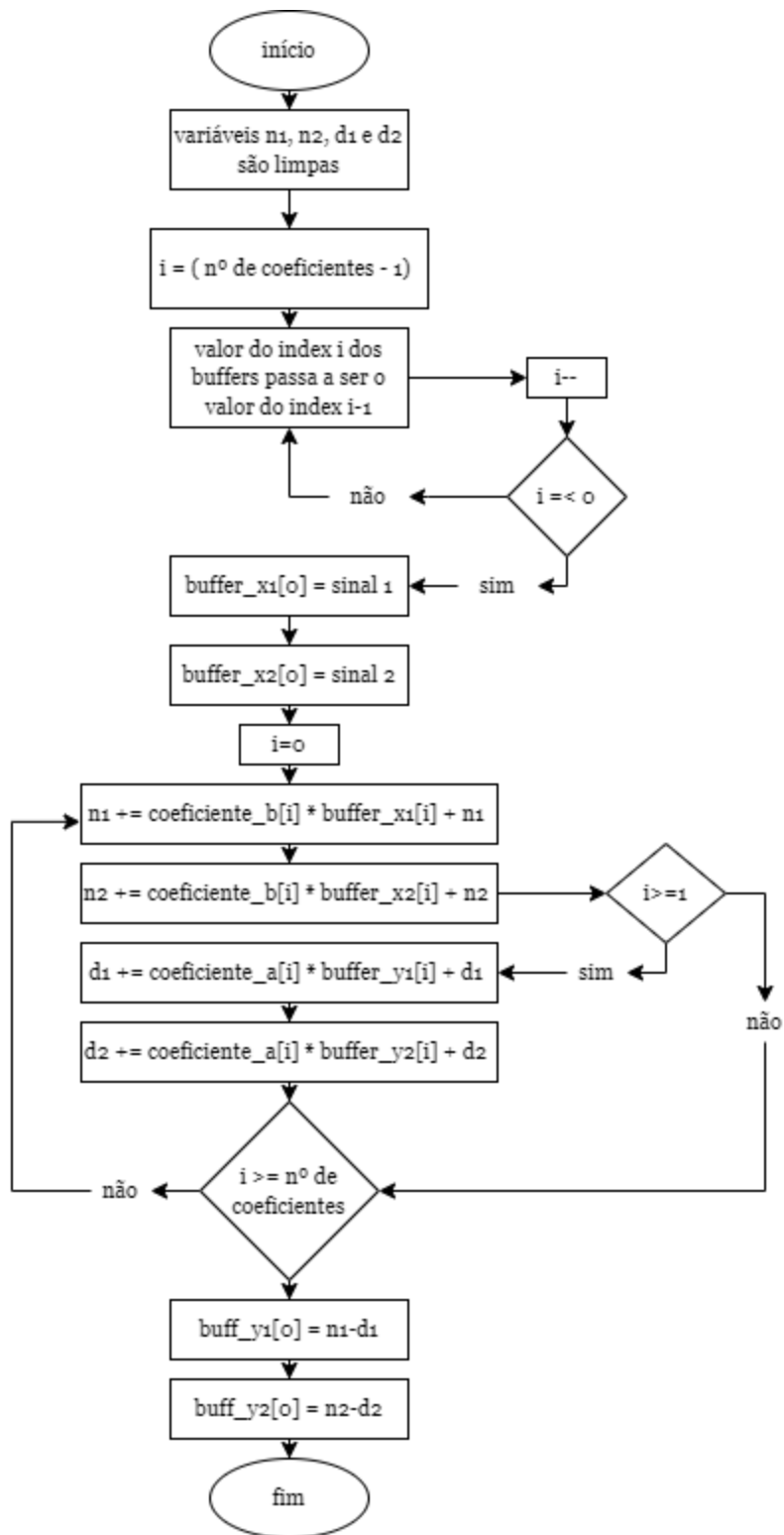


Figura 92 - Fluxograma da função "Filter\_IIR".

### 5.2.2. Função ângulo

Para o cálculo do ângulo foram também construídas duas funções. Uma delas apenas com o propósito de inicialização, que tem como argumento um ponteiro para uma estrutura, onde será guardada a média de ambos os sinais, as constantes para o método de cálculo do ângulo elétrico (explicado mais à frente), a posição inicial, o número de ciclos elétricos, uma variável que pretende guardar o valor do desfasamento dos filtros, uma matriz de duas posições para o ângulo elétrico (posição atual e posição anterior) e uma variável para o ângulo mecânico. A função de inicialização (“*angleStructInit*”) tem como propósito guardar a média de cada sinal, o valor das constantes para o cálculo do ângulo e o desfasamento de ângulo causado pelos filtros (que neste caso será  $1^\circ$ ). Esta função deve ser chamada uma única vez.

A função para o cálculo do ângulo (“*angle\_calc*”) tem como argumentos de entrada um ponteiro para uma estrutura do mesmo tipo e duas variáveis *double* para cada sinal convertido. Esta função deve ser chamada em cada ciclo de conversão, após o processamento da função de filtragem ter terminado. Primeiramente, a função subtrai a média a cada sinal e, de seguida, efetua o cálculo do ângulo elétrico pelo método que se explicará a seguir. Este valor é guardado na posição zero da matriz para o propósito.

Esta função pressupõe que o arranque do motor é apenas feito após a conversão de 1000 amostras. Serve isto para dar tempo à resultante da filtragem estabilizar e ter-se uma referência de posição inicial. Sendo assim, enquanto o número de conversões for inferior a zero, o ângulo mecânico não é calculado e pressupõe-se que é zero, uma vez que o motor não se moveu e, por isso, a estrutura não saiu da posição inicial. Devido ao tempo de estabilização do filtro, apenas se contabilizam as conversões a partir da 200 para a determinação da posição inicial. Após 1000 conversões é feita a média dos últimos 800 valores obtidos pelo cálculo do ângulo elétrico para se poder subtrair a posição inicial do ciclo elétrico ao ângulo mecânico.

Uma vez que o cálculo do ângulo elétrico através do arcotangente apenas permite medir o ângulo dentro de um ciclo elétrico (que neste caso corresponderá a  $1,8^\circ$ ), é necessário contabilizar o número de ciclos e multiplicá-los por  $1,8^\circ$  para o cálculo do ângulo mecânico. Sendo assim, o algoritmo verifica se o ângulo elétrico atual é inferior a  $-3,135$  radianos ( $-\pi$  é o ângulo inicial do ciclo quando a rotação do motor é no sentido horário) e se o ângulo elétrico anterior é superior a  $3,135$  ( $\pi$  é o valor de ângulo final do ciclo quando a rotação do motor é no sentido horário). Se esta condição se verificar é

incrementado o valor 1 à variável “ciclo”. O mesmo processo é feito no sentido inverso, sentido anti-horário, em que caso se verifiquem as condições é decrementado 1 à variável ciclo. Por fim, é calculado o ângulo mecânico conforme a Equação 39. Tanto o ângulo elétrico, como os ciclos são convertidos em ângulos mecânicos que depois lhes é subtraída a posição inicial, para se saber a posição atual da estrutura relativamente à posição inicial. A variável do ângulo elétrico utilizada é a posição zero da matriz correspondente. Por último, o valor do index zero da matriz do ângulo elétrico é passado para o index 1, para no próximo cálculo se saber a posição anterior, se é necessário incrementar/decrementar o contador de ciclos ou não.

$$\theta = \left( \frac{\alpha - \text{posição inicial}}{2 * \pi} + \text{ciclo} \right) * 1,8 \quad (37)$$

Em relação ao cálculo do ângulo elétrico, o método tradicional é através do arcotangente de meio ciclo ou ciclo completo, no entanto esta função exige algum tempo de processamento. Para diminuir este tempo de cálculo no microcontrolador verificou-se a viabilidade da fórmula em (fastatan2, n.d., <http://www.f3.to/portfolio/math/fastatan2.htm>).

Através do osciloscópio verificou-se o tempo de processamento da função representada na Figura 93, com a função “atan2” e o novo método de cálculo do ângulo. Verificou-se que com o novo método, o tempo de processamento reduziu de 2,8µs para 2,32µs (Figura 94 e Figura 95). De seguida, simulou-se em Matlab o cálculo com estas duas funções, subtraindo-se uma à outra para se verificar o desvio da segunda.

Nota: nesta simulação ambas as funções foram aplicadas as duas ondas sinusoidais perfeitas com uma frequência de 20Hz e com 100000 amostras por segundo. O resultado entre a diferença dos métodos está representado na Figura 96 e na Figura 97. A diferença máxima obtida é de 0,2684° por ciclo. Dada a diferença ser bastante reduzida optou-se por este novo método para o cálculo do ângulo elétrico.

Os tempos de processamento medidos na Figura 77 já têm em conta este novo método.

Este método exige primeiro que se verifique se o módulo do sinal 1 é maior que o módulo do sinal 2 e caso se verifique o cálculo é feito pela Equação 39. Caso não se verifique o cálculo é feito pela Equação 40.

$$\alpha = \frac{-c2 * \text{sinal1} * \text{sinal2}}{(\text{sinal1})^2 + c1 * (\text{sinal2})^2} + c3 * \frac{\text{sinal1}}{|\text{sinal1}|} \quad (38)$$

$$\alpha = -c3 * \frac{\text{sin}a1 * \text{sin}a2}{|\text{sin}a1 * \text{sin}a2|} + c3 * \frac{\text{sin}a1}{|\text{sin}a1|} + \frac{c2 * \text{sin}a1 * \text{sin}a2}{(\text{sin}a2)^2 * (\text{sin}a1)^2 * c1} \quad (39)$$

Valores das constantes para o resultado em radianos é:

- $c1=0,28088$ ;
- $c2=1$ ;
- $c3 = c2 * \pi * 0,5$ ;

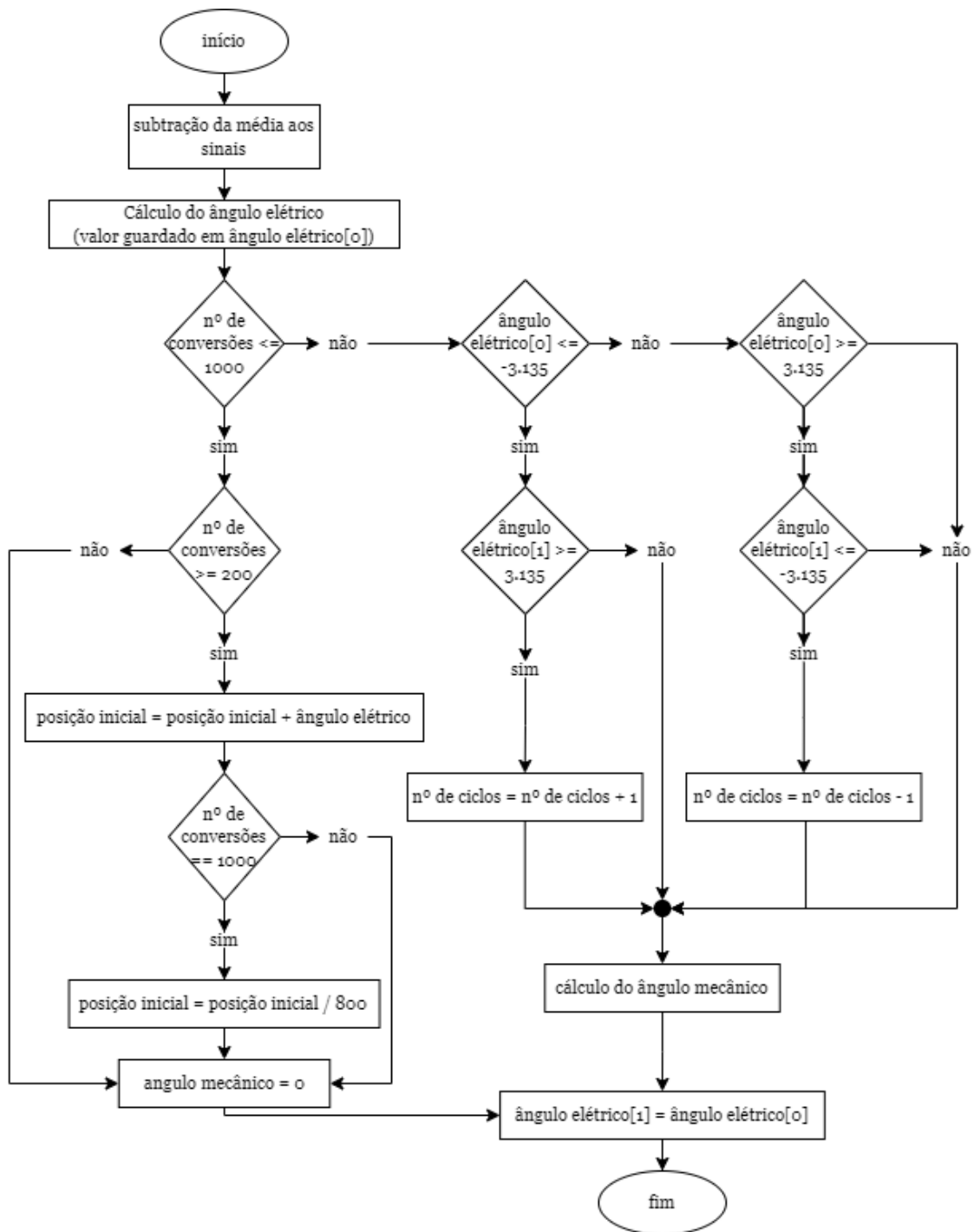


Figura 93 – Fluxograma do algoritmo do cálculo do ângulo.

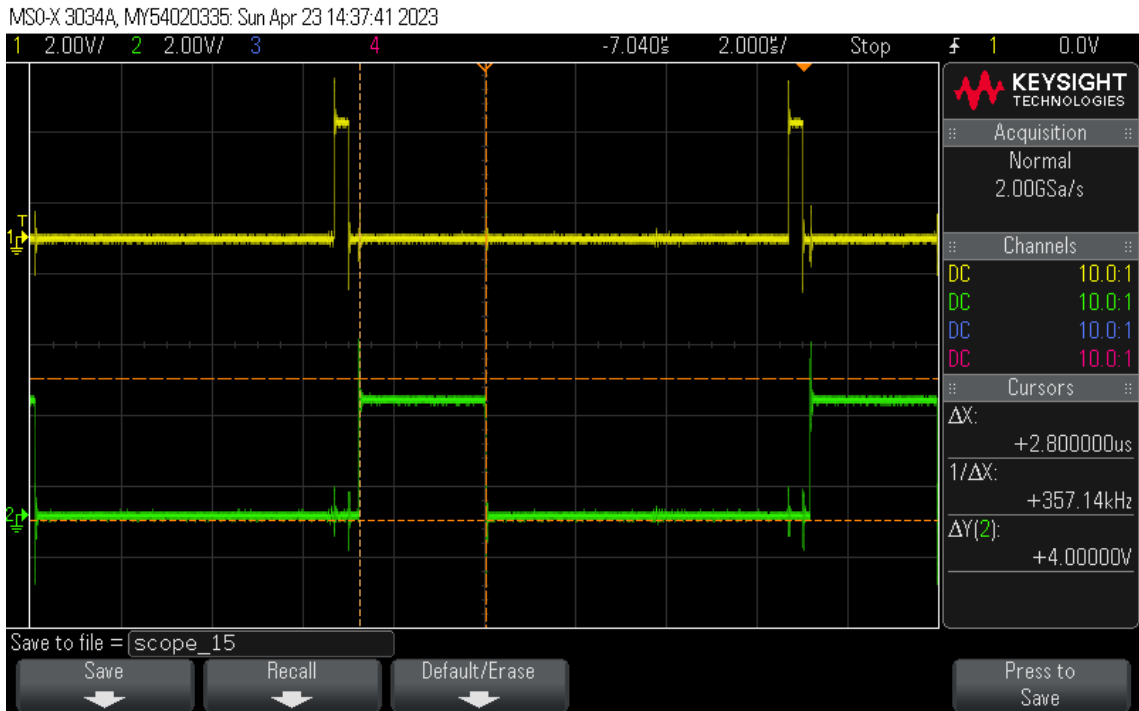


Figura 94 – Tempo de processamento da função *angle\_calc* utilizando atan2 para o cálculo do ângulo elétrico.

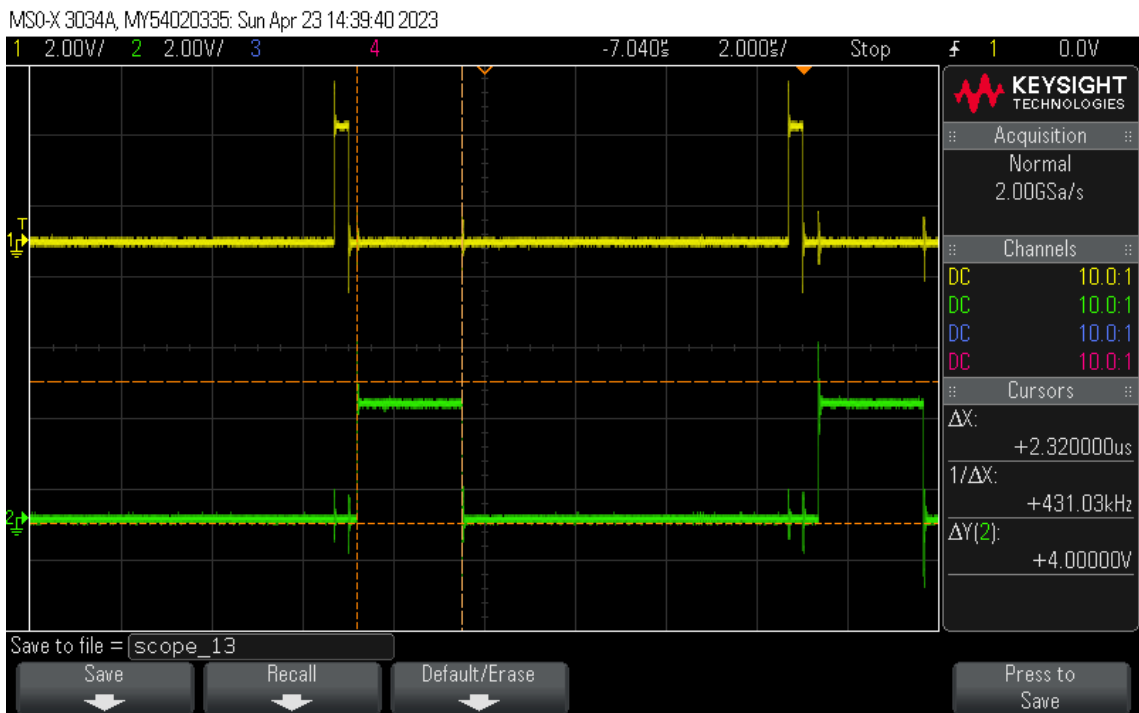


Figura 95 - Tempo de processamento da função *angle\_calc* utilizando nova fórmula.

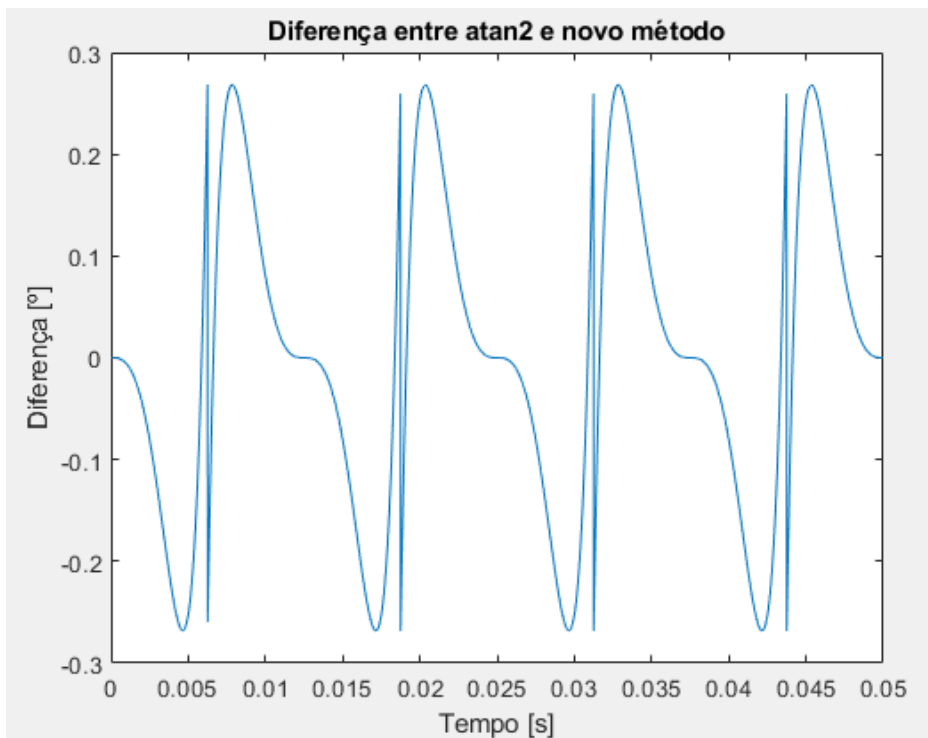


Figura 96 – Diferença entre cálculo pela fórmula atan2 e nova fórmula.

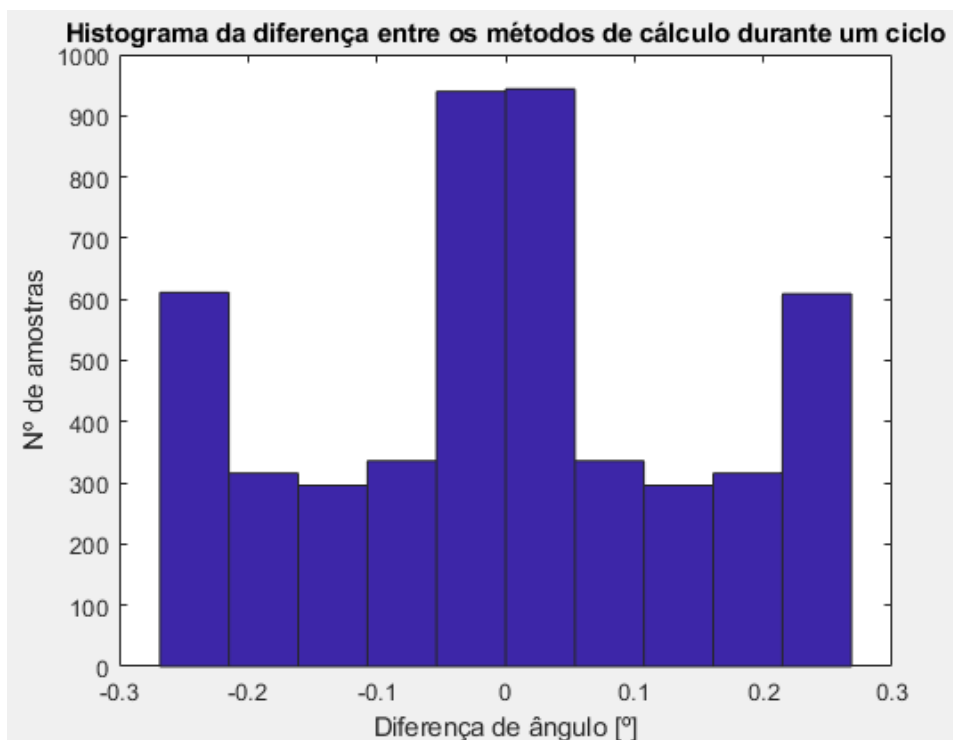


Figura 97 – Histograma da diferença entre cálculo pela fórmula atan2 e nova fórmula.

### 5.2.3. Funcionamento do 2º teste

O funcionamento do 2º teste é similar ao funcionamento do 1º teste, sendo que na inicialização são agora chamadas as funções “*IIRStructInit*” e “*angleStructInit*” com os valores que correspondentes a cada condição. Já o funcionamento da máquina de estados tem ligeiras diferenças, uma vez que agora após o tempo de inicialização, se passa para a aquisição de dados e só é iniciado o movimento do motor após 1000 ciclos de conversão, isto é, após se converterem 1000 dados de cada sinal. Após se converterem 25000 dados de cada sinal passa-se para o estado de envio através da UART. No estado de aquisição de dados, assim que é feita a conversão analógica para digital dos dois sinais, estes dados passam pela função “*Filter\_IIR*” e os resultados desta passam pela função “*angle\_calc*”. Numa primeira etapa os resultados da função “*Filter\_IIR*” serão guardados nas matrizes de onde se enviarão os dados para se analisar os dados filtrados. Posteriormente serão os resultados do cálculo do ângulo elétrico e do ângulo mecânico que serão enviados através da UART. Uma vez que os dados resultantes do filtro e do cálculo do ângulo são do tipo *double*, por questões de memória do microcontrolador só se guardarão amostras de 25000 dados quando se pretender retirar amostras dos dois sinais filtrados e 50000, quando se pretender retirar o resultado do cálculo do ângulo elétrico e mecânico.

Dado que o envio via UART passa a ser de dados tipo *double* foi necessário alterar a função “*senddata*”. Para isso cada dado a ser enviado é convertido em *string* e de seguida é enviado cada caracter da *string* através do registo U2TXREG. Uma vez que o *serialplot* (*software* que recebe os dados da UART) permite fazer a conversão de novo de *string* para *double*.

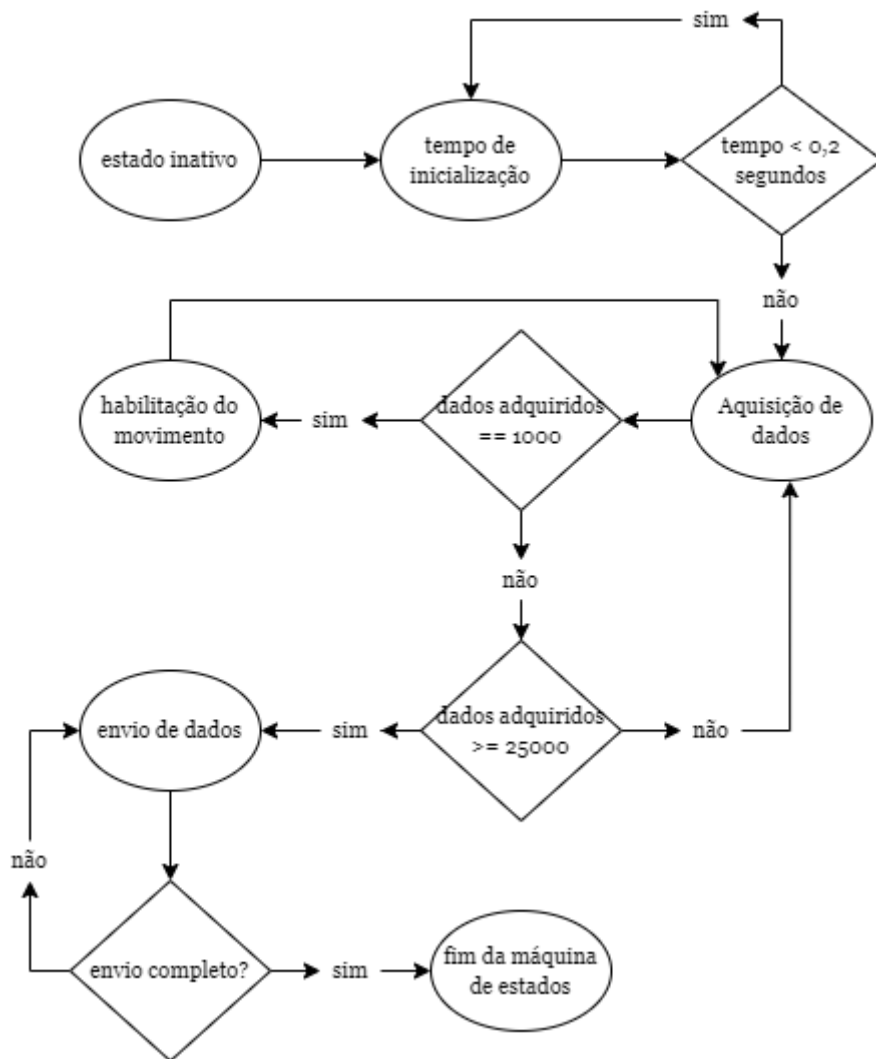


Figura 98 – Funcionamento do 2º teste.

#### 5.2.4. Resultados

Como mencionado anteriormente, o objetivo deste teste prendia-se por verificar o funcionamento dos algoritmos do filtro e do cálculo do ângulo no código do microcontrolador. Primeiramente enviaram-se os dados dos sinais filtrados do microcontrolador para o computador para se analisar no domínio da frequência o comportamento destes sinais. Foi feita uma comparação com os dados anteriormente retirados e filtrados em Matlab. Como se pode verificar nas Figuras 99, 100, 101 e 102 a amostra filtrada pelo microcontrolador tem um comportamento semelhante à filtrada no Matlab o que demonstra que a função do filtro está a funcionar como se pretende.

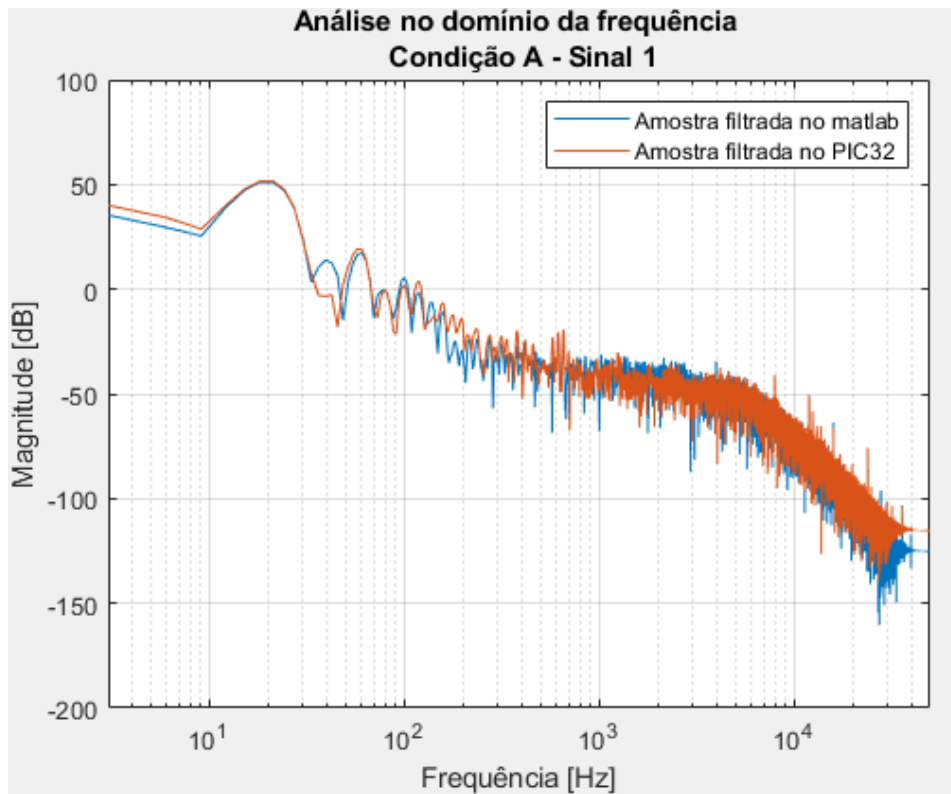


Figura 99 – Magnitude de frequências da condição A sinal 1.

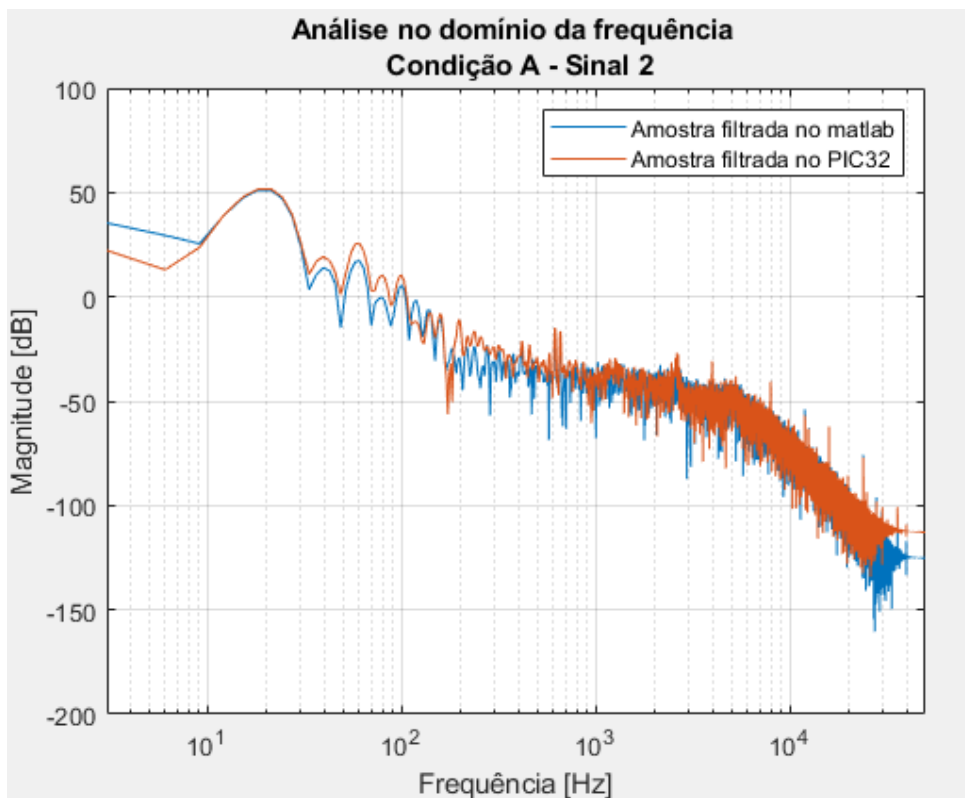


Figura 100 - Magnitude de frequências da condição A sinal 2.

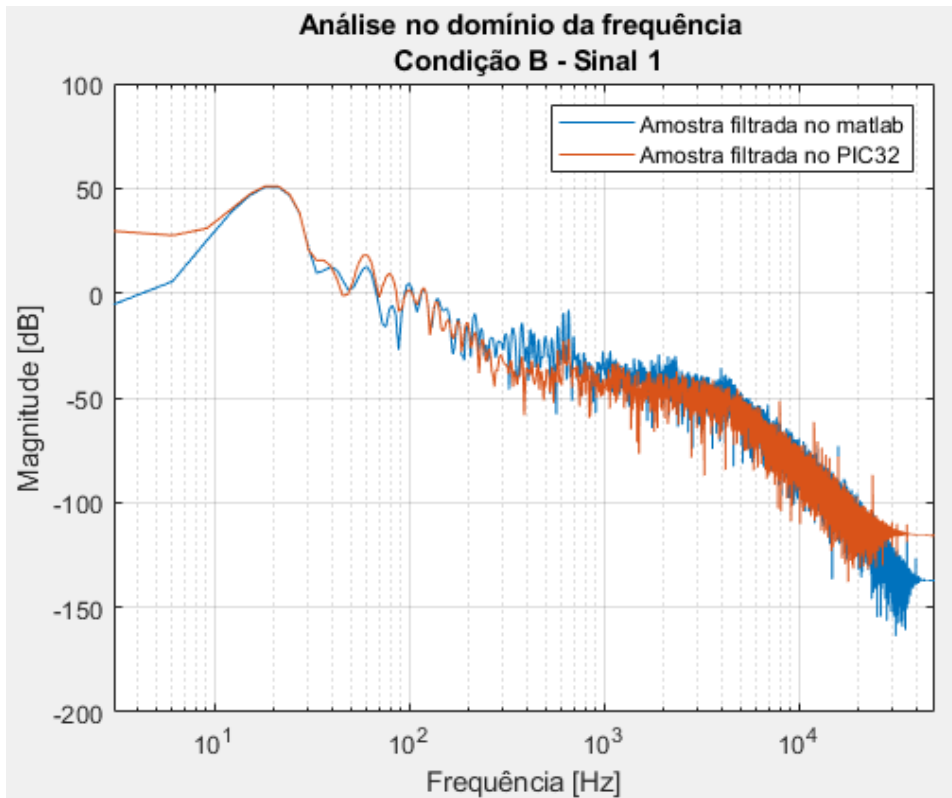


Figura 101 - Magnitude de frequências da condição B Sinal 1.

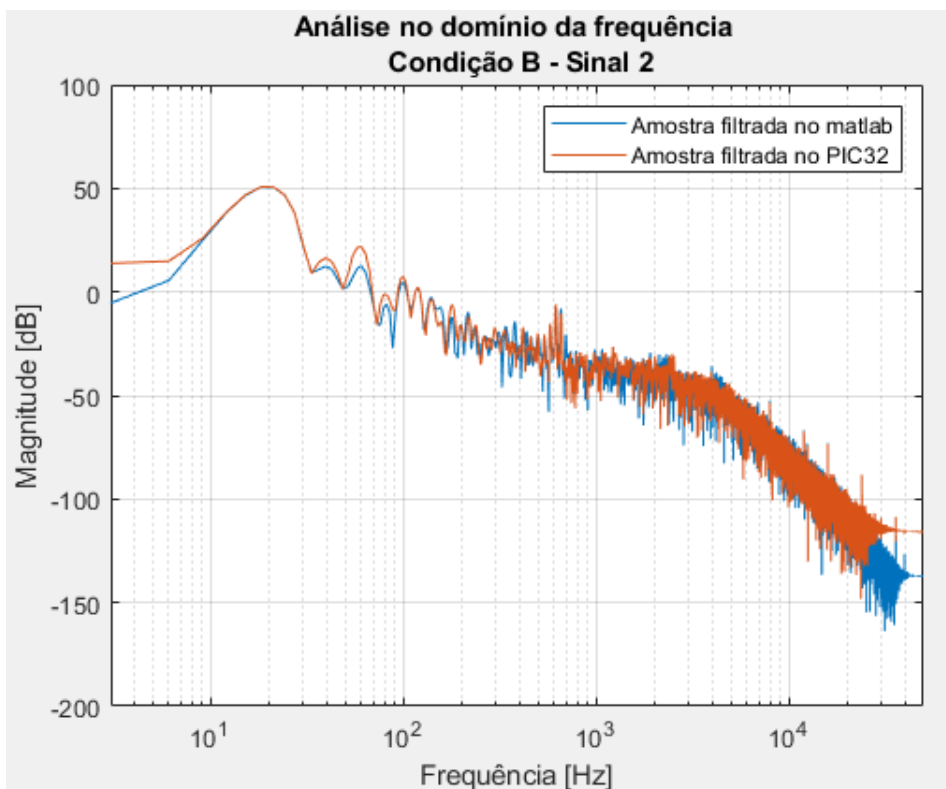


Figura 102 - Magnitude de frequências da condição B Sinal 2.

Na Figura 103 e na Figura 104 poderá ver-se a resultante do cálculo do ângulo elétrico mantendo formas ligeiramente irregulares.

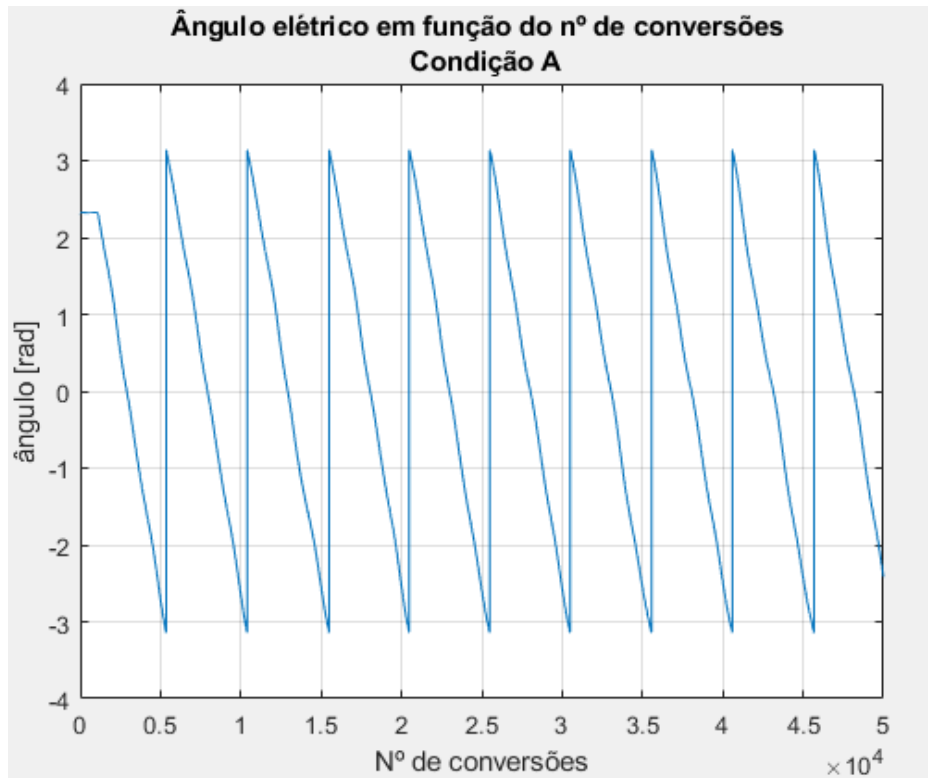


Figura 103 – Ângulo elétrico da condição A.

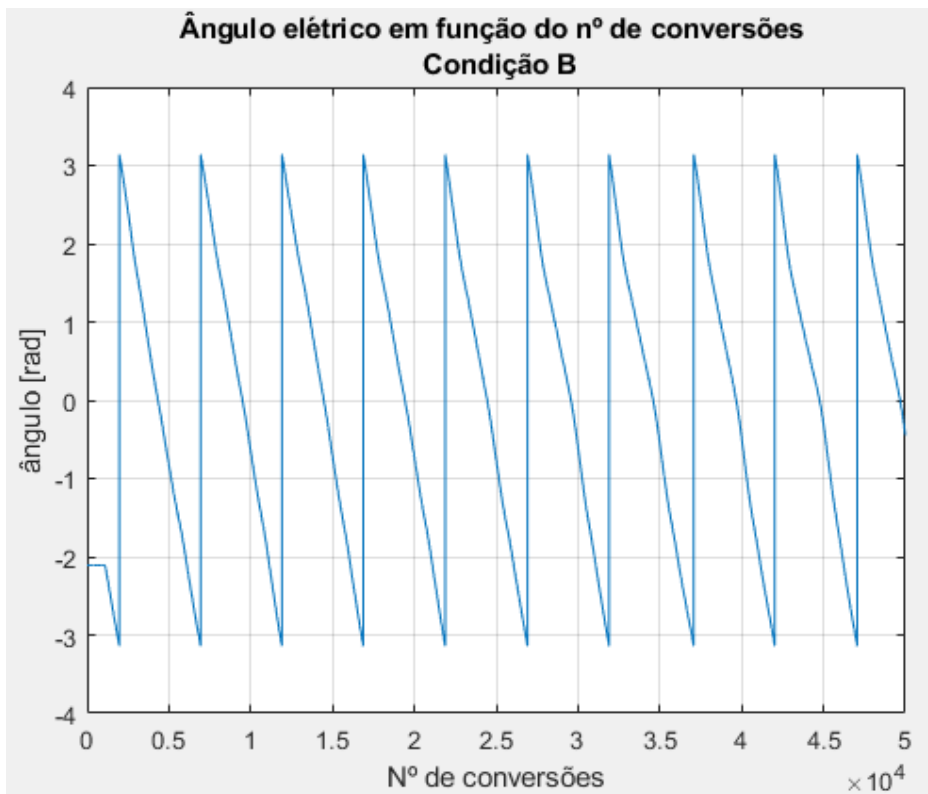


Figura 104 - Ângulo elétrico da condição B.

Na Figura 105 e na Figura 106 poderá observar-se a posição angular em cada amostra recolhida. Dado que a velocidade angular conferida foi de  $36^\circ/\text{s}$  e as amostras recolhidas são referentes 0,5s (50000 amostras retiradas a 100kHz), sendo que o motor só inicia movimento ao fim de 1000 amostras, seria de esperar que o motor tivesse uma posição angular final de  $17,64^\circ$  ( $49\text{k} \cdot 36^\circ / 100\text{k}$ ). No entanto, a posição angular da condição A é de  $17,1411^\circ$  e da condição B de  $17,5240^\circ$ . Este ligeiro desfasamento entre a posição esperada e a posição alcançada pode dever-se tanto a erros na obtenção do sinal (descentramento dos sinais em relação ao eixo, amplitudes diferentes entre sinais, furos do disco não uniformes) como a prisões do motor.

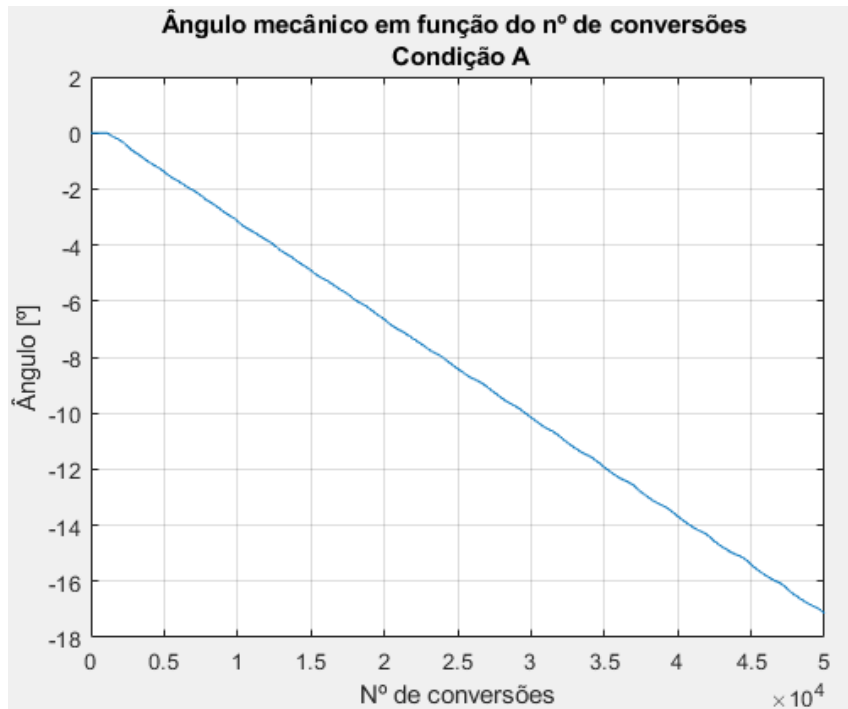


Figura 105 - Ângulo mecânico da condição A.

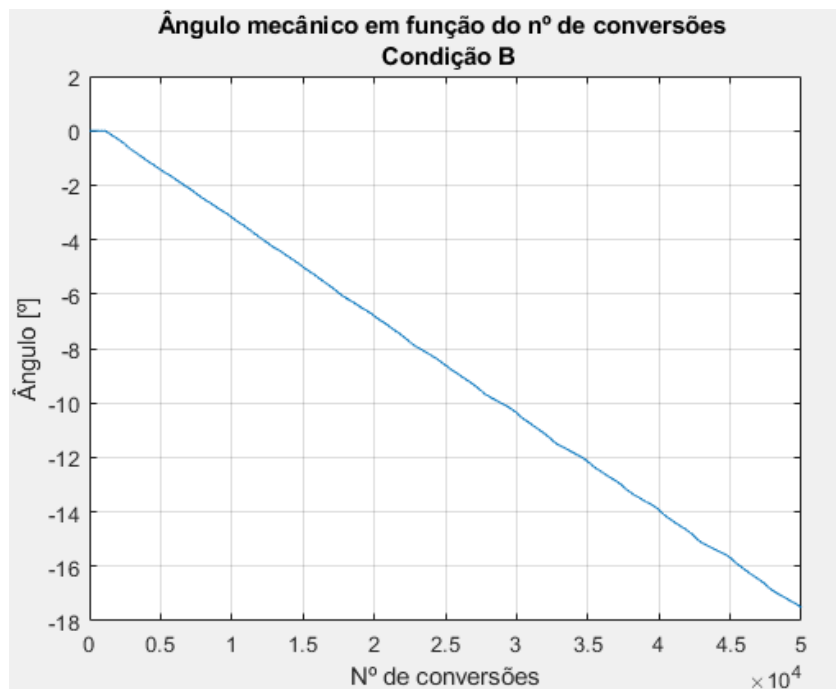


Figura 106 - Ângulo mecânico da condição B.

Por fim pode-se verificar na Figura 107 que a condição B, também neste teste, reduz de forma ligeiramente mais eficaz o ruído de alta frequência.

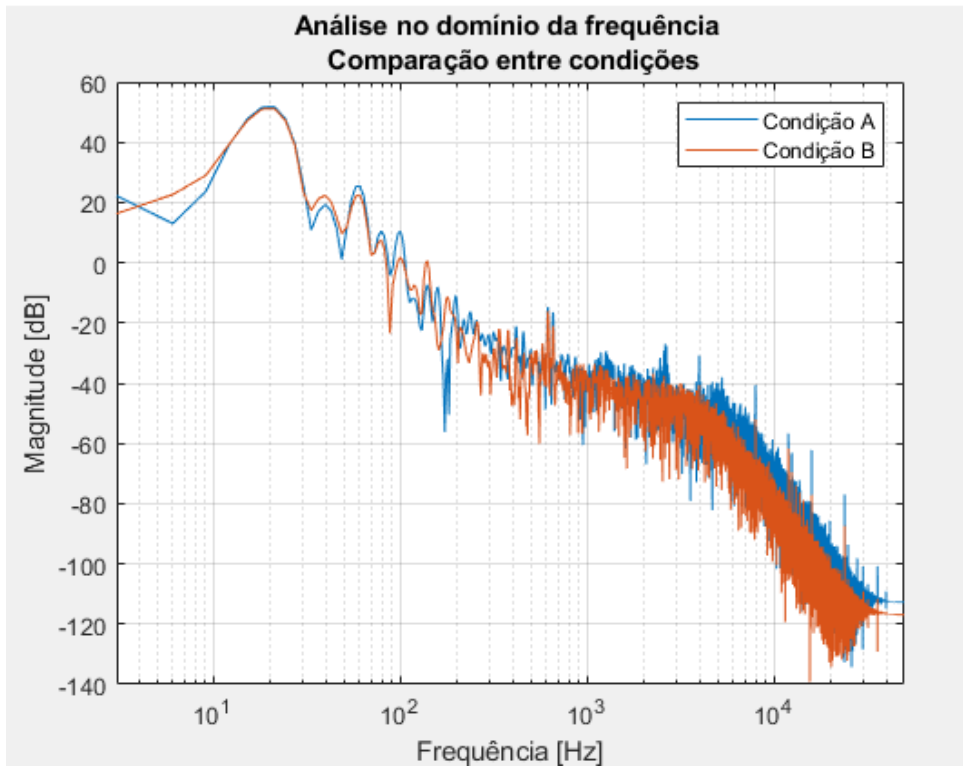


Figura 107 – Comparação da magnitude das frequências entre a condição A e a condição B.

## 6. Conclusões

### 6.1. Conclusões Gerais

O desenvolvimento do presente codificador ótico envolveu múltiplas áreas de engenharia como a eletrônica, processamento de sinal, programação e modulação tridimensional.

O objetivo prendeu-se por criar um codificador ótico de alta resolução e respetivo algoritmo para um microcontrolador PIC32. Para isto foi desenhada uma placa de circuito impresso com dois LEDs e dois fototransístores, impressas peças em PLA, incluindo um disco codificador, peça de suporte para os LEDs e fototransístores, bem como uma estrutura que conferiu movimento de rotação à peça de suporte em relação ao disco codificador, através de um motor de passo. Através de uma placa de circuito impresso, com um microcontrolador PIC32, controlou-se o movimento do motor e fez-se a aquisição e processamento de sinal. O microcontrolador está conectado a um *driver* de motor de passo colocado numa *breadboard* de forma a controlar o motor. Nesta *breadboard* foram também colocados filtros analógicos de passa-baixo de forma a reduzir ruído de alta frequência antes dos sinais serem adquiridos e processados pelo microcontrolador.

A meta foi otimizar ao máximo a resolução e atenuar o quanto possível o ruído indesejado de forma a melhorar a exatidão do codificador ótico. Primeiramente foram definidas condições iniciais como frequência do ciclo elétrico (20Hz), frequência de conversão (100kHz) de forma a não se perder resolução e desfasamento máximo causado pelos filtros analógico e digital (1 grau elétrico). Foram dimensionados dois filtros analógicos (com diferentes frequências de corte), desenvolvidos os algoritmos e efetuado o primeiro teste.

Durante o primeiro teste verificou-se que o disco codificador tinha irregularidades significantivas pelo que deveria ser reimpresso até se obterem dados aceitáveis. Após o primeiro teste, com um disco codificador com menos irregularidades, verificaram-se dois problemas: o valor médio dos sinais ser diferente entre os dois e também diferente de zero e ruído de alta frequência algo significativo.

O primeiro problema foi atenuado pela subtração do valor médio em ambos os sinais. Não se optou por um filtro de passa-alto pelo desfasamento que causaria no sinal e tempo

de processamento que necessitaria. No entanto, através de simulações em Matlab verificou-se que a solução utilizada continuaria a permitir um desvio significativo, em relação ao eixo, em alguns ciclos.

O segundo problema foi atenuado por filtros digitais IIR de passa-baixo. Para o dimensionamento dos filtros digitais, através de um osciloscópio foi medido o tempo de processamento que sobrava em cada ciclo de conversão (100kHz) e medido o tempo de processamento de filtros com diferentes números de coeficientes. Chegou-se à conclusão que o filtro não poderia ter mais de 5 coeficientes pelo que foram dimensionados dois filtros IIR com 5 coeficientes de forma que o desfasamento somado aos filtros analógicos não ultrapassasse a condição de desfasamento máximo de 1º elétrico.

O segundo teste foi feito já com a subtração da média e filtro digital. Neste teste pode-se verificar o funcionamento esperado do algoritmo do filtro digital e que a redução de ruído de alta frequência é ligeiramente maior na condição em que o filtro digital tem uma menor frequência de corte (e o filtro analógico uma maior frequência de corte).

Em conclusão, encontraram-se algumas limitações neste projeto:

- A falta de um método de medição do ângulo mecânico não permitiu otimizar a frequência de conversão em relação ao número de coeficientes do filtro digital e frequência de corte para ambos os filtros;
- O mesmo sinal ter flutuações da média ao longo dos 360º do disco codificador fez com que a subtração da média não reduzisse significativamente o desvio do sinal em relação ao eixo;
- A calibração analógica dos sinais permitiu diferentes amplitudes entre os dois sinais que provocaram, certamente, inexatidões no cálculo do ângulo;
- As tensões de referência não permitiram um usufruto ótimo da resolução do conversor analógico digital. A resolução teórica máxima possível com os materiais utilizados seria de  $4,396e-4^\circ$  ( $1,8^\circ/4095$ ). Considerando a amplitude média do sinal de menor amplitude, a resolução média atingida foi de  $6,818e-4^\circ$  ( $1,6^\circ/2640$ ), uma vez que a amplitude média desse sinal é de 2640.

## 6.2. Sugestões de Trabalho Futuro

Uma placa de circuito impresso dedicada ao sistema poderia atenuar em parte estes problemas. A diferença de amplitude entre os dois sinais poderia ser reduzida através do controlo da intensidade de corrente nos LEDs do codificador. Esta intensidade de corrente poderá ser controlada digitalmente através de potenciômetros digitais manipulados através de comunicação SPI (recurso disponível no microcontrolador PIC32) ou através do recurso DAC (*Digital-to-Analog Converter*) que permite definir a tensão de saída de determinados pinos do microcontrolador com uma resolução de 4095, em função das tensões de referência que podem ser internas ou externas. Poderá também redefinir-se os valores tensão de referência externa de forma a aumentar-se a resolução do codificador. Uma vez que o valor mínimo obtido foi de -1667 que corresponde a 0,2325V, poder-se-ia utilizar uma tensão de referência de 0,2V. O desenho desta placa possibilitaria ainda a utilização de filtros analógicos ativos, permitindo uma maior redução do ruído de alta frequência. De mencionar que uma placa com todos os recursos integrados concederia um melhor aspeto estético e mais fácil integração na estrutura, uma vez que não seria necessário a utilização de placas externas como a *breadboard*.

De forma a estabilizar-se o valor médio seria necessária uma impressão mais cuidada e de maior precisão do disco codificador de forma que os furos tenham menos irregularidades possíveis e maior uniformidade entre eles.

Para trabalhos futuros poderia ser útil a criação de um sistema que pudesse medir a exatidão do sistema de posicionamento. Um sistema deste género, para além do benefício óbvio de tornar mensurável a qualidade do codificador, poderia permitir a otimização da relação entre frequência de conversão e número de coeficientes no filtro digital, bem como otimização da escolha da frequência de corte dos filtros em relação ao desfasamento causado ao sinal.



# Referências Bibliográficas

- 3D Printer Universe. (n.d.). *ARTILLERY SIDEWINDER X1 3D PRINTER*. Retrieved May 20, 2023, from <https://3dprinteruniverse.com/products/artillery-sidewinder-x1-3d-printer>
- Ahmed, S. N. (2015). Essential statistics for data analysis. In *Physics and Engineering of Radiation Detection* (pp. 541–593). Elsevier. <https://doi.org/10.1016/b978-0-12-801363-2.00009-7>
- ALL3DP, & Jackson O’Connel. (n.d.). *PLA Plastic/Material – The Ultimate Guide*. Retrieved May 20, 2023, from <https://all3dp.com/2/what-is-pla-plastic-material-properties/>
- Allan H. Robbins, & Wilhelm C. Miller. (2012). *Allan H. Robbins, Wilhelm C Miller - Circuit Analysis\_ Theory and Practice-Cengage Learning (2012)*.
- Allan H. Robbins, & Wilhelm C. Miller. (2013). *Circuit Analysis - Theory and Practice* (Cengage Learning, Ed.; 5th Edition).
- Allegro Microsystems. (2022). *A4988 Datasheet*. [www.allegromicro.com](http://www.allegromicro.com)
- Asadi, F. (2023). *Applied Numerical Analysis with MATLAB®/Simulink®*.
- Burke J., Moynihan J. F., & Unterkofler K. (2000). Extraction of High Resolution Position Information From Sinusoidal Encoders. *Analog Devices*.
- Chen, Y., & Dong, F. (2013). Robot machining: Recent development and future research issues. *International Journal of Advanced Manufacturing Technology*, 66(9–12), 1489–1497. <https://doi.org/10.1007/s00170-012-4433-4>
- Colao, J. (2020). Fast Reacting, Optical Encoder Feedback System for Miniature Motor Driven Applications. *Analog Devices*, 54.
- Davis, N. (2017). *An Introduction to Filters*. <https://www.allaboutcircuits.com/technical-articles/an-introduction-to-filters/>
- Eco, J., & Limjoco, A. (2016). *Ferrite Beads Demystified*.

*fastatan2*. (n.d.). Retrieved May 13, 2023, from <http://www.f3.to/portfolio/math/fastatan2.htm>

Fleming, A. J. (2013). A review of nanometer resolution position sensors: Operation and performance. *Sensors and Actuators, A: Physical*, 190, 106–126. <https://doi.org/10.1016/j.sna.2012.10.016>

Fraden, J. (2004). *Handbook of Modern Sensors, Physics, Designs, and Applications*. Springer.

Gao, W. (2010). *Precision Nanometrology*. Springer.

Hao, S., Liu, Y., & Hao, M. (2009). Study on a Novel Absolute Magnetic Encoder. *IEEE International Conference on Robotics and Biomimetics*.

Hasan Yavuz Özderya. (n.d.). *Serial Plot Hackaday*. Retrieved May 20, 2022, from <https://hackaday.io/project/5334-serialplot-realtime-plotting-software>

Hentai Motor. (n.d.). *Hybrid Stepper Motor Datasheet-42BYGHM Series-15A-JB*.

Hoang, H. Van, & Jeon, J. W. (2007). Signal Compensation and Extraction of High Resolution Position for Sinusoidal Magnetic Encoders. *International Conference on Control, Automation and Systems*.

Hoang, H. Van, & Jeon, J. W. (2011). An efficient approach to correct the signals and generate high-resolution quadrature pulses for magnetic encoders. *IEEE Transactions on Industrial Electronics*, 58(8), 3634–3646. <https://doi.org/10.1109/TIE.2010.2093472>

International Federation of Robotics. (2022, October). *World Robotics Report: “All-Time High” with Half a Million Robots Installed in one Year*. <https://ifr.org/ifr-press-releases/news/wr-report-all-time-high-with-half-a-million-robots-installed>

Kendall Su. (2002). *ANALOG FILTERS, SECOND EDITION* (Kluwer Academic Publishers, Ed.; Second).

Kester, W. (2005). *Data Conversion Handbook*.

Kester, W. (2009a). *Decoupling Techniques*. <http://www.fair-rite.com>

- Kester, W. (2009b). Optical Encoders. *Analog Devices, MT-029 TUTORIAL*.
- Kester, W., & Analog Devices. (2003). *Mixed-Signal and DSP Design Techniques*. Elsevier Science.
- Kicad EDA. (2021). *Introduction KiCad*. <https://www.kicad.org/help/report-an-issue/>
- Kim, S. H., Nam, E., Ha, T. I., Hwang, S. H., Lee, J. H., Park, S. H., & Min, B. K. (2019). Robotic Machining: A Review of Recent Progress. In *International Journal of Precision Engineering and Manufacturing* (Vol. 20, Issue 9, pp. 1629–1642). SpringerOpen. <https://doi.org/10.1007/s12541-019-00187-w>
- Kochis, G., & Haddington Dynamics. (2015, January 25). *Stepper Opto Feedback*. [https://github.com/HaddingtonDynamics/Dexter/blob/Stable\\_2020\\_02\\_04\\_Co neDrive/Hardware/Opto/09011-00122-0.pdf](https://github.com/HaddingtonDynamics/Dexter/blob/Stable_2020_02_04_Co neDrive/Hardware/Opto/09011-00122-0.pdf)
- LITE-ON Technology Corporation. (n.d.-a). *IR Emitter and Detector Product Data Sheet LTE-4206*. <http://www.liteon.com/opto>
- LITE-ON Technology Corporation. (n.d.-b). *IR Emitter and Detector Product Data Sheet LTE-4206E*. <http://www.liteon.com/opto>
- Lynch, K. M., Marchuk, N., & Elwin, M. L. (2016). Sensors. In *Embedded Computing and Mechatronics with the PIC32* (pp. 317–340). Elsevier. <https://doi.org/10.1016/B978-0-12-420165-1.00021-4>
- Lyons, R. (2008). *Quadrature Signals: Complex, But Not Complicated*.
- Lyons, R. G. (2011). *Understanding Digital Signal Processing* (Third Edition). Pearson Education, Inc.
- Malvino, A. P., & Bates, D. J. (2016). *Electronic principles* (8th ed.). McGraw-Hill Education.
- Mathworks. (n.d.). *Introduction to Filter Designer*. Retrieved March 25, 2023, from <https://www.mathworks.com/help/signal/ug/introduction-to-filter-designer.html>
- MathWorks. (n.d.-a). *MathWorks filter 1-D digital filter*. Retrieved March 20, 2023, from <https://www.mathworks.com/help/matlab/ref/filter.html>

- MathWorks. (n.d.-b). *MathWorks Filter Designer*. Retrieved April 20, 2023, from <https://www.mathworks.com/help/dsp/ug/using-filter-designer.html>
- MathWorks. (n.d.-c). *MathWorks FFT Fast Fourier Transform*. Retrieved April 20, 2023, from <https://www.mathworks.com/help/matlab/ref/fft.html>
- Microchip Technology Inc. (2011). *PIC32 Family Reference Manual Section 16. Output Compare*. <http://www.microchip.com>
- Microchip Technology Inc. (2012). *PIC32 Family Reference Manual Section 21. UART*. <http://www.microchip.com>
- Microchip Technology Inc. (2013a). *PIC32 Family Reference Manual Section 14. Timers*. <http://www.microchip.com>
- Microchip Technology Inc. (2013b). *PIC32 Family Reference Manual Section 42. Oscillators with Enhanced PLL*. <http://www.microchip.com>
- Microchip Technology Inc. (2015a). *PIC32 Family Reference Manual Section 8. Interrupts*. <http://www.microchip.com>
- Microchip Technology Inc. (2015b). *PIC32 Family Reference Manual Section 12. I/O Ports*. <http://www.microchip.com>
- Microchip Technology Inc. (2019). *PIC32 Family Reference Manual Section 22. 12-bit High-Speed Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC)*. <http://www.microchip.com>
- Microchip Technology Inc. (2020). *MPLAB® PICkit™ 4 In-Circuit Debugger User's Guide*.
- Microchip Technology Inc. (2021). *PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Family Operating Conditions*.
- Microchip Technology Inc. (2022). *MPLAB X IDE User's Guide*.
- My3DConcepts. (n.d.). *HOW 3D PRINTING WORKS?* Retrieved May 20, 2022, from <http://my3dconcepts.com/explore/how-3d-printing-works/>

- Newton, J. (2021, May 25). *Dexter Hardware*.  
<https://github.com/HaddingtonDynamics/Dexter/wiki/Hardware>
- Newton, J., & Haddington Dynamics. (2020, August 26). *CAD Dexter 1*.  
<https://github.com/HaddingtonDynamics/Dexter/tree/master/Hardware>
- PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Family Operating Conditions*. (n.d.).
- Precision PCB Services. (n.d.). *Guide to PCB Soldering Techniques*. Retrieved January 20, 2023, from <https://www.pcb-repair.com/resources/pcb-soldering-techniques/>
- PTRobotics. (n.d.). *A4988 Stepper Motor Driver Carrier*. Retrieved January 22, 2023, from <https://www.ptrobotics.com/modulos-motor-pontes-h/1822-a4988-stepper-motor-driver-carrier.html>
- Reis, E., Gralha, C., & Monteiro, M. P. (2022). Surveying communities of users of MATLAB and clone languages. *Journal of Computer Languages*, 73.  
<https://doi.org/10.1016/j.cola.2022.101170>
- Solidworks. (2015). *INTRODUCING SOLIDWORKS*.
- Solomon, I. J., Sevel, P., & Gunasekaran, J. (2020). A review on the various processing parameters in FDM. *Materials Today: Proceedings*, 37(Part 2), 509–514.  
<https://doi.org/10.1016/j.matpr.2020.05.484>
- Tayo Yuden. (n.d.). *BK2125HS470-T Specs*. Retrieved June 1, 2023, from <https://ds.yuden.co.jp/TYCOMPAS/ut/detail?pn=BK2125HS470-T%20%20&u=M>
- Texas Instruments Incorporated. (2019). How to Implement Comparators for Improving Performance of Rotary Encoders in Industrial Drive Applications. *Texas Instruments Incorporated, SNOAA41*. [www.ti.com](http://www.ti.com)
- Watanabe, T., Kon, M., Nabeshima, N., & Taniguchi, K. (2014). An angle encoder for super-high resolution and super-high accuracy using SelfA. *Measurement Science and Technology*, 25(6). <https://doi.org/10.1088/0957-0233/25/6/065002>

- Ye, S. H., Wang, Y., Ren, Y. J., & Li, D. K. (2006). Robot calibration using iteration and differential kinematics. *Journal of Physics: Conference Series*, 48(1), 1–6.  
<https://doi.org/10.1088/1742-6596/48/1/001>
- Yu, H., Chen, X., Liu, C., Cai, G., & Wang, W. (2021a). A survey on the grating based optical position encoder. *Optics & Laser Technology*, 143, 107352.  
<https://doi.org/10.1016/j.optlastec.2021.107352>
- Yu, H., Chen, X., Liu, C., Cai, G., & Wang, W. (2021b). A survey on the grating based optical position encoder. *Optics & Laser Technology*, 143, 107352.  
<https://doi.org/10.1016/j.optlastec.2021.107352>
- Zheng, D., Zhang, S., Wang, S., Hu, C., & Zhao, X. (2015). A capacitive rotary encoder based on quadrature modulation and demodulation. *IEEE Transactions on Instrumentation and Measurement*, 64(1), 143–153.  
<https://doi.org/10.1109/TIM.2014.2328456>
- Zumbahlen, H. (2008). *Linear Circuit Design Handbook*.

## Anexos

Tabela A1. Relação entre as fontes de relógio e atributos do microcontrolador. Adaptado de (Microchip Technology Inc., 2013b).

	ADC	UART	Timers	Comparador de saída	Interrupções
SOSC			X		
FRC	X				
SYSCLK	X				X
PBCLK1					
PBCLK2		X			
PBCLK3	X		X	X	
REFCLKO3	X				

Tabela A2. Registo PBxDIV. Adaptado de (Microchip Technology Inc., 2021)

<b>PBxDIV</b>			
<b>Posição (bit)</b>	<b>Subregistro</b>	<b>Valores a atribuir</b>	<b>Função</b>
[16, 31]	Não Implementado		
15	ON	1	PBxCLK desabilitado
		0	PBxCLK habilitado
[7, 14]	...	...	...
[0, 6]	PBDIV	127	$PBCLK_x = SYSCLK/128$
		126	$PBCLK_x = SYSCLK/127$
		...	...
		4	$PBCLK_x = SYSCLK/5$
		3	$PBCLK_x = SYSCLK/4$
		2	$PBCLK_x = SYSCLK/3$
		1	$PBCLK_x = SYSCLK/2$
		0	$PBCLK_x = SYSCLK/1$

Tabela A3. Registro T1CON. Adaptado de (Microchip Technology Inc., 2021)

T1CON				
Posição (bit)	Subregistro	Valores a atribuir	Função	Nota
[16 , 31]	Não Implementado	Ler como "0"		
15	ON	1	Timer é habilitado	
		0	Timer é desabilitado	
14	Não Implementado	Ler como "0"		
[6,14]	...		...	
[5 , 4]	TCKPS	3	valor de pré-escala 1:256	
		2	valor de pré-escala 1:64	
		1	valor de pré-escala 1:8	
		0	valor de pré-escala 1:1	
3	Não Implementado	Ler como "0"		
2	TSYNC	1	A entrada de relógio externa é sincronizada	Quando TCS = 0, bit ignorado
		0	A entrada de relógio externa não é sincronizada	
1	TCS	1	Utilizado relógio externo do pino TxCKI	Quando TCS = 0, bit ignorado
		0	Utilizado relógio interno	
0	Não Implementado	Ler como "0"		

Tabela A4. Registo TxCON. Adaptado de (Microchip Technology Inc., 2021)

TxCON				
Posição (bit)	Subregisto	Valores a atribuir	Função	Nota
[16, 31]	Não Implementado	Ler como "0"		
15	ON	1	<i>Timer</i> é habilitado	
		0	<i>Timer</i> é desabilitado	
14	Não Implementado	Ler como "0"		
[7, 13]	...		...	
[6, 4]	TCKPS	7	valor de pré-escala 1:256	
		6	valor de pré-escala 1:64	
		5	valor de pré-escala 1:32	
		4	valor de pré-escala 1:16	
		3	valor de pré-escala 1:8	
		2	valor de pré-escala 1:4	
		1	valor de pré-escala 1:2	
		0	valor de pré-escala 1:1	
3	T32	1	TMRx e TMRy formam um <i>timer</i> de 32 bits	
		0	Utiliza apenas um <i>timer</i> como <i>timer</i> de 16 bits	
2	Não Implementado	0		
1	TCS	1	Utilizado relógio externo do pino TxCK1	
		0	Utilizado relógio interno	
0	Não Implementado	Ler como "0"		

Tabela A5. Vetores e Registos das rotinas de serviço e interrupção de cada Timer. Adaptado de (Microchip Technology Inc., 2021)

	<b>Vetor</b>	<b>Flag</b>		<b>Habilitação</b>	
		<b>Registo</b>	<b>Subregisto</b>	<b>Registo</b>	<b>Subregisto</b>
<i>Timer 1</i>	<i>_TIMER_1_VECTOR</i>	IFS0	T1IF	IECo	T1IE
<i>Timer 2</i>	<i>_TIMER_2_VECTOR</i>	IFS0	T2IF	IECo	T2IE
<i>Timer 3</i>	<i>_TIMER_3_VECTOR</i>	IFS0	T3IF	IECo	T3IE
<i>Timer 4</i>	<i>_TIMER_4_VECTOR</i>	IFS0	T4IF	IECo	T4IE
<i>Timer 5</i>	<i>_TIMER_5_VECTOR</i>	IFS0	T5IF	IECo	T5IE
<i>Timer 6</i>	<i>_TIMER_6_VECTOR</i>	IFS0	T6IF	IECo	T6IE
<i>Timer 7</i>	<i>_TIMER_7_VECTOR</i>	IFS1	T7IF	IFS1	T7IE
<i>Timer 8</i>	<i>_TIMER_8_VECTOR</i>	IFS1	T8IF	IFS1	T8IE
<i>Timer 9</i>	<i>_TIMER_9_VECTOR</i>	IFS1	T9IF	IFS1	T9IE

Tabela A6. Registos das rotinas de serviço e interrupção de cada Timer. Adaptado de (Microchip Technology Inc., 2021)

	<b>Priorização</b>		<b>Subpriorização</b>	
	<b>Registo</b>	<b>Subregisto</b>	<b>Registo</b>	<b>Subregisto</b>
<i>Timer 1</i>	IPC1	T1IP	IPC1	T1IS
<i>Timer 2</i>	IPC2	T2IP	IPC2	T2IS
<i>Timer 3</i>	IPC3	T3IP	IPC3	T3IS
<i>Timer 4</i>	IPC4	T4IP	IPC4	T4IS
<i>Timer 5</i>	IPC6	T5IP	IPC6	T5IS
<i>Timer 6</i>	IPC7	T6IP	IPC7	T6IS
<i>Timer 7</i>	IPC8	T7IP	IPC8	T7IS
<i>Timer 8</i>	IPC9	T8IP	IPC9	T8IS
<i>Timer 9</i>	IPC10	T9IP	IPC10	T9IS

Tabela A7. Relação entre os módulos do Comparador de saída e os Módulos Timer. Adaptado de (Microchip Technology Inc., 2021)

Módulo Comparador de saída	Timerx	Timery	
OC1	<i>Timer2</i>	<i>Timer3</i>	CFGCONbits.OCACLK = 0
...	<i>Timer2</i>	<i>Timer3</i>	
OC9	<i>Timer2</i>	<i>Timer3</i>	
OC1	<i>Timer4</i>	<i>Timer5</i>	CFGCONbits.OCACLK = 1
OC2	<i>Timer4</i>	<i>Timer5</i>	
OC3	<i>Timer4</i>	<i>Timer5</i>	
OC4	<i>Timer2</i>	<i>Timer3</i>	
OC5	<i>Timer2</i>	<i>Timer3</i>	
OC6	<i>Timer2</i>	<i>Timer3</i>	
OC7	<i>Timer6</i>	<i>Timer7</i>	
OC8	<i>Timer6</i>	<i>Timer7</i>	
OC9	<i>Timer6</i>	<i>Timer7</i>	

Tabela A8. Registo OCxCON. Adaptado de (Microchip Technology Inc., 2021)

OCxCON			
Posição (bit)	Subregisto	Valores a atribuir	Função
[16, 31]	Não Implementado	0	
15	ON	1	Comparador de saída é desabilitado
		0	Comparador de saída é habilitado
14	Não Implementado	0	
13	SIDL	1	Interrompe a operação quando o dispositivo entra em modo desativo ( <i>Idle Mode</i> )
		0	Continua a operação quando o dispositivo entra em modo desativo ( <i>Idle Mode</i> )
[6, 12]	Não Implementado	1	Escrita no registo TMR1 são ignoradas até que a escrita pendente esteja completa
		0	Gravações consecutivas estão ativadas
5	OC32	1	Modo 32 bits é ativado
		0	Modo 16 bits é ativado
4	OCFLT	1	Ocorreu uma condição de falha do PWM
		0	Não ocorreu nenhuma condição de falha do PWM
3	OCTSEL	1	Timery é a fonte de clock para este módulo de comparação de saída
		0	Timerx é a fonte de clock para este módulo de comparação de saída
[0, 2]	OCM	7	Modo PWM ligado; Pino de falha habilitado
		6	Modo PWM ligado; Pino de falha desabilitado
		5	Modo de comparação dupla: pulso contínuo
		4	Modo de comparação dupla: pulso único
		3	Modo de comparação única: nível alternado
		2	Modo de comparação única: nível baixo
		1	Modo de comparação única: nível alto
		0	Nenhum modo ativo

Tabela A9. Registro OCxCON. Adaptado de (Microchip Technology Inc., 2021)

	Vetor	Flag		Habilitação	
		Registro	Subregistro	Registro	Subregistro
OC1	_OUTPUT_COMPARE_1_VECTOR	IFS0	OC1IF	IEC0	OC1IE
OC2	_OUTPUT_COMPARE_2_VECTOR	IFS0	OC2IF	IEC0	OC2IE
OC3	_OUTPUT_COMPARE_3_VECTOR	IFS0	OC3IF	IEC0	OC3IE
OC4	_OUTPUT_COMPARE_4_VECTOR	IFS0	OC4IF	IEC0	OC4IE
OC5	_OUTPUT_COMPARE_5_VECTOR	IFS0	OC5IF	IEC0	OC5IE
OC6	_OUTPUT_COMPARE_6_VECTOR	IFS0	OC6IF	IEC0	OC6IE
OC7	_OUTPUT_COMPARE_7_VECTOR	IFS1	OC7IF	IFS1	OC7IE
OC8	_OUTPUT_COMPARE_8_VECTOR	IFS1	OC8IF	IFS1	OC8IE
OC9	_OUTPUT_COMPARE_9_VECTOR	IFS1	OC9IF	IFS1	OC9IE

Tabela A10. Registros das rotinas de serviço e interrupção de cada comparador de saída. Adaptado de (Microchip Technology Inc., 2021)

	Priorização		Subpriorização	
	Registro	Subregistro	Registro	Subregistro
OC1	IPC1	OC1IP	IPC1	OC1IS
OC2	IPC3	OC2IP	IPC3	OC2IS
OC3	IPC4	OC3IP	IPC4	OC3IS
OC4	IPC5	OC4IP	IPC5	OC4IS
OC5	IPC6	OC5IP	IPC6	OC5IS
OC6	IPC7	OC6IP	IPC7	OC6IS
OC7	IPC8	OC7IP	IPC8	OC7IS
OC8	IPC9	OC8IP	IPC9	OC8IS
OC9	IPC10	OC9IP	IPC10	OC9IS

Tabela A11. Registro UxMODE. Adpatado de (Microchip Technology Inc., 2021)

<b>UxMODE</b>			
<b>Posição (bit)</b>	<b>Subregistro</b>	<b>Valores a atribuir</b>	<b>Função</b>
[16 , 31]	Não Implementado	Ler como "0"	
15	ON	1	UARTx é ativado
		0	UARTx é desativado
14	Não Implementado	Ler como "0"	
[4 , 13]	...		...
3	BRGH	1	Divisor de Clock = 4
		0	Divisor de Clock = 16
[1, 2]	PDSEL	3	Dados de 9 bits sem paridade
		2	Dados de 8 bits com paridade ímpar
		1	Dados de 8 bits com paridade par
		0	Dados de 8 bits sem paridade
0	STESSEL	1	2 bits de paragem
		0	1 bit de paragem

Tabela A12. Registo UxSTA da posição 6 à posição 31. Adaptado de (Microchip Technology Inc., 2021)

UxSTA				
Posição (bit)	Subregisto	Valores a atribuir	Função	Nota
[25 , 31]	Não Implementado	Ler como "0"		
[16 , 24]	...		...	
[14, 15]	UTXISEL	3	Reservado. Não utilizar	
		2	Interrupção é gerada enquanto o buffer de transmissão estiver vazio	
		1	Interrupção é gerada quando todos os caracteres tiverem sido transmitidos	
		0	Interrupção é gerada enquanto o buffer de transmissão tiver pelo menos um espaço vazio	
13	...		...	
12	URXEN	1	Recetor UARTx é ativado e pino de receção é controlado pela UARTx	
		0	Recetor UARTx é desativado	
11	...		...	
10	UTXEN	1	Transmissor UARTx é ativado e pino de transmissão é controlado pela UARTx	
		0	Transmissor UARTx é desativado	
9	UTXBF	1	Buffer de Transmissão está cheio	Apenas de leitura
		0	Buffer de Transmissão ainda tem pelo menos um espaço por preencher	
8	TMRT	1	o shifter de transmissão está vazio e o buffer de transmissão está vazio	Apenas de leitura
		0	o shifter de transmissão não está vazio e a transmissão está a ocorrer ou na fila	
[6, 7]	URXISEL	3	Reservado. Não utilizar	
		2	O bit sinalizador de interrupção é ativado quando o buffer de receção está 3/4 ou mais cheio	
		1	O bit sinalizador de interrupção é ativado quando o buffer de receção está 1/2 ou mais cheio	
		0	O bit sinalizador de interrupção é ativado quando o buffer de receção não está vazio	

Tabela A13. Registo UxSTA da posição 6 à posição 31. Adaptado de (Microchip Technology Inc., 2021)

UxSTA				
Posição (bit)	Subregisto	Valores a atribuir	Função	Nota
[4, 5]	...		...	
3	PERR	1	Encontrado erro de paridade no caracter atual	Apenas de leitura
		0	Não foi encontrado erro de paridade	
2	FERR	1	Encontrado erro de enquadramento no caracter atual	Apenas de leitura
		0	Não foi encontrado erro de enquadramento	
1	OERR	1	Buffer de receção transbordou	
		0	Buffer de receção não transbordou	
0	URXDA	1	Buffer de receção tem dados	Apenas de leitura
		0	Buffer de receção está vazio	

Tabela A14. Vetores e Registos das rotinas de serviço e interrupção de cada UART. Adpatado de (Microchip Technology Inc., 2021)

	Vetor	Flag		Habilitação	
		Registo	Subregisto	Registo	Subregisto
UART1	_UART1TX_VECTOR	IFS3	U1TXIF	IEC3	U1TXIE
	_UART1_RX_VECTOR	IFS3	U1RXIF	IEC3	U1RXIF
	_UART1_FAULT_VECTOR	IFS3	U1EIF	IEC3	U1EIF
UART2	_UART2TX_VECTOR	IFS4	U2TXIF	IEC4	U2TXIE
	_UART2_RX_VECTOR	IFS4	U2RXIF	IEC4	U2RXIF
	_UART2_FAULT_VECTOR	IFS4	U2EIF	IEC4	U2EIF
UART3	_UART3TX_VECTOR	IFS4	U3TXIF	IEC4	U3TXIE
	_UART3_RX_VECTOR	IFS4	U3RXIF	IEC4	U3RXIF
	_UART3_FAULT_VECTOR	IFS4	U3EIF	IEC4	U3EIF
UART4	_UART4TX_VECTOR	IFS5	U4TXIF	IEC5	U4TXIE
	_UART4_RX_VECTOR	IFS5	U4RXIF	IEC5	U4RXIF
	_UART4_FAULT_VECTOR	IFS5	U4EIF	IEC5	U4EIF
UART5	_UART5TX_VECTOR	IFS5	U5TXIF	IEC5	U5TXIE
	_UART5_RX_VECTOR	IFS5	U5RXIF	IEC5	U5RXIF
	_UART5_FAULT_VECTOR	IFS5	U5EIF	IEC5	U5EIF
UART6	_UART6TX_VECTOR	IFS5	U6TXIF	IEC5	U6TXIE
	_UART6_RX_VECTOR	IFS5	U6RXIF	IEC5	U6RXIF
	_UART6_FAULT_VECTOR	IFS5	U6EIF	IEC5	U6EIF

Tabela A15. Registos das rotinas de serviço e interrupção de cada UART. Adpatado de (Microchip Technology Inc., 2021)

	Priorização		Subpriorização	
	Registo	Subregisto	Registo	Subregisto
UART1	IPC28	U1TXIF	IPC28	U1TXIE
	IPC28	U1RXIF	IPC28	U1RXIF
	IPC28	U1EIF	IPC28	U1EIF
UART2	IPC36	U2TXIF	IPC36	U2TXIE
	IPC36	U2RXIF	IPC36	U2RXIF
	IPC36	U2EIF	IPC36	U2EIF
UART3	IPC39	U3TXIF	IPC39	U3TXIE
	IPC39	U3RXIF	IPC39	U3RXIF
	IPC39	U3EIF	IPC39	U3EIF
UART4	IPC42	U4TXIF	IPC42	U4TXIE
	IPC42	U4RXIF	IPC42	U4RXIF
	IPC43	U4EIF	IPC43	U4EIF
UART5	IPC45	U5TXIF	IPC45	U5TXIE
	IPC45	U5RXIF	IPC45	U5RXIF
	IPC45	U5EIF	IPC45	U5EIF
UART6	IPC47	U6TXIF	IPC47	U6TXIE
	IPC47	U6RXIF	IPC47	U6RXIF
	IPC47	U6EIF	IPC47	U6EIF

Tabela A16. Registo ADCCON1. Adpatado de (Microchip Technology Inc., n.d.)

ADCCON1			
Posição (bit)	Subregisto	Valores a atribuir	Função
[24, 31]	...		...
23	FRACT	1	Valores convertidos em fracionário
		0	Valores convertidos em inteiro
22	SELRES	3	12 bits
		2	10 bits
		1	8 bits
		0	6 bits
[16, 21]	...		...
15	ON	1	ADC é habilitado
		0	ADC é desabilitado
[0, 14]	...		...

Tabela A17. Registro ADCCON2. Adpatado de(Microchip Technology Inc., 2021)

<b>ADCCON2</b>			
<b>Posição (bit)</b>	<b>Subregistro</b>	<b>Valores a atribuir</b>	<b>Função</b>
31	BGVRRDY	1	tensão de intervalo de banda e tensão de referência preparadas
		0	tensão de intervalo de banda e tensão de referência não estão preparadas
30	REFFLT	1	tensão de referência apresenta problemas
		0	tensão de referência não apresenta problemas
29	...		...
[26, 28]	CVDCPL	7	17,5pF
		6	15pF
		5	12,5pF
		4	10pF
		3	7,5pF
		2	5pF
		1	2,5pF
		0	0pF
[16, 25]	SAMC	1023	1025*TAD7
		...	...
		1	3*TAD7
		0	2*TAD7
[7, 15]	...		...
[0, 6]	ADCDIV	127	254 * TQ = TAD7
		...	...
		3	6*TQ = TAD7
		2	4*TQ = TAD7
		1	2*TQ = TAD7
		0	Reservado

Tabela A18. Registro ADCCON3. Adaptado de (Microchip Technology Inc., 2021)

<b>ADCCON3</b>			
<b>Posição (bit)</b>	<b>Subregistro</b>	<b>Valores a atribuir</b>	<b>Função</b>
[30, 31]	ADCSEL	3	FRC
		2	REFCLK3
		1	System Clock (Tcy)
		0	PBCLK3
[24, 29]	CONCLKDIV	63	64* TCLK = TQ
		...	...
		3	4* TCLK = TQ
		2	3 * TCLK = TQ
		1	2 * TCLK = TQ
23	DIGEN7	1	ADC7 é desabilitado digitalmente
		0	ADC7 é habilitado digitalmente
[21, 22]	...		...
20	DIGEN4	1	ADC4 é desabilitado digitalmente
		0	ADC4 é habilitado digitalmente
19	DIGEN3	1	ADC3 é desabilitado digitalmente
		0	ADC3 é habilitado digitalmente
18	DIGEN2	1	ADC2 é desabilitado digitalmente
		0	ADC2 é habilitado digitalmente
17	DIGEN1	1	ADC1 é desabilitado digitalmente
		0	ADC1 é habilitado digitalmente
16	DIGENo	1	ADCo é desabilitado digitalmente
		0	ADCo é habilitado digitalmente
[13, 15]	VREFSEL	[4,7]	Reservado
		3	ADref+ = External VRefh
			ADref- = External Vrefl
		2	ADref+ = AVDD
			ADref- = External Vrefl
		1	ADref+ = External Vrefh
ADref- = Avss			
0	ADref+ = AVDD		
	ADref- = Avss		
[8, 12]	...		...
7	GLSWTRG	1	gatilho de conversão ativado
		0	gatilho de conversão desativado
6	GSWTRG	1	gatilho de conversão ativado no momento em que lhe é atribuído este valor
		0	gatilho de conversão desativado
[0, 5]	...		...

Tabela A19. Registro ADCCON3. Adaptado de (Microchip Technology Inc., 2021)

<b>ADCxTIME</b>			
<b>Posição (bit)</b>	<b>Subregistro</b>	<b>Valores a atribuir</b>	<b>Função</b>
[29, 31]	...		...
[24, 25]	SELRES	3	12 bits
		2	10 bits
		1	8 bits
		0	6 bits
23	Não implementado		
[16, 22]	ADCDIV	127	254 * TQ = TADx
		...	...
		3	6*TQ = TADx
		2	4*TQ = TADx
		1	2*TQ = TADx
		0	Reservado
[10, 15]	Não implementado		
[0, 9]	SAMC	1023	1025 * TADx
		....	...
		3	5 * TADx
		2	4 * TADx
		1	3 * TADx
		0	2 * TADx

Tabela A20. Módulos ADC e respectivas entradas analógicas. Adaptado de (Microchip Technology Inc., 2021)

<b>Módulo</b>	<b>Entrada positiva</b>	<b>Entrada de referência</b>
ADC <sub>0</sub>	AN <sub>0</sub>	AN <sub>5</sub>
	AN <sub>45</sub>	VREFL
ADC <sub>1</sub>	AN <sub>1</sub>	AN <sub>6</sub>
	AN <sub>46</sub>	VREFL
ADC <sub>2</sub>	AN <sub>2</sub>	AN <sub>7</sub>
	AN <sub>47</sub>	VREFL
ADC <sub>3</sub>	AN <sub>3</sub>	AN <sub>8</sub>
	AN <sub>48</sub>	VREFL
ADC <sub>4</sub>	AN <sub>4</sub>	AN <sub>9</sub>
	AN <sub>49</sub>	VREFL
ADC <sub>7</sub>	AN <sub>5</sub>	AN <sub>10</sub>
	AN <sub>6</sub>	VREFL
	...	
	AN <sub>42</sub>	

Tabela A21. Valores e funções dos subregistros TRGSRCx. Adaptado de (Microchip Technology Inc., 2021)

<b>Subregistro</b>	<b>Valores a atribuir</b>	<b>Função</b>
TRGSRCx	[13, 31]	Reservado
	12	Comparador 2
	11	Comparador 1
	10	OCMP <sub>5</sub>
	9	OCMP <sub>3</sub>
	8	OCMP <sub>1</sub>
	7	TMR <sub>5</sub>
	6	TMR <sub>3</sub>
	5	TMR <sub>1</sub>
	4	INT <sub>0</sub> interrupção externa
	3	STRIG
	2	GLSWTRG
	1	GSWTRG
	0	Sem Gatilho

Tabela A22. Registro ADCTRGMODE desde a posição 16 à posição 31. Adaptado de(Microchip Technology Inc., 2021)

<b>ADCTRGMODE</b>			
<b>Posição (bit)</b>	<b>Subregistro</b>	<b>Valores a atribuir</b>	<b>Função</b>
[26, 31]	Não Implementado		
[24, 25]	SH4ALT	3	Reservado
		2	Reservado
		1	AN49
		0	AN4
[22, 23]	SH3ALT	3	Reservado
		2	Reservado
		1	AN48
		0	AN3
[20, 21]	SH2ALT	3	Reservado
		2	Reservado
		1	AN47
		0	AN2
[19, 18]	SH1ALT	3	Reservado
		2	Reservado
		1	AN46
		0	AN1
[16, 17]	SHoALT	3	Reservado
		2	Reservado
		1	AN45
		0	AN4

Tabela A23. Subregistro WKUPCLKCNT. Adaptado de (Microchip Technology Inc., 2021)

<b>Subregistro</b>	<b>Valores a atribuir</b>	<b>Função</b>
WKUPCLKCNT	15	32768 ciclos de relógio
	14	16384 ciclos de relógio
	13	8192 ciclos de relógio
	12	4096 ciclos de relógio
	11	2048 ciclos de relógio
	10	1024 ciclos de relógio
	9	512 ciclos de relógio
	8	256 ciclos de relógio
	7	128 ciclos de relógio
	6	64 ciclos de relógio
	5	32 ciclos de relógio
	4	16 ciclos de relógio
	3	16 ciclos de relógio
	2	16 ciclos de relógio
	1	16 ciclos de relógio
	0	16 ciclos de relógio

Tabela A24. Vetor e Registos das rotinas de serviço e interrupção global do ADC. Adpatado de (Microchip Technology Inc., 2021)

<b>Vetor</b>	<b>Flag</b>		<b>Habilitação</b>		<b>Priorização</b>		<b>Subpriorização</b>	
	<b>Registro</b>	<b>Subregistro</b>	<b>Registro</b>	<b>Subregistro</b>	<b>Registro</b>	<b>Subregistro</b>	<b>Registro</b>	<b>Subregistro</b>
ADC_VECTOR_	IFS1	ADCIF	IEC1	ADCIE	IPC11	ADCIP	IPC11	ADCIS