

Controlo por Movimento Ocular

João Matos, Maria Carvalho, Maria Loureiro, Faculdade de Engenharia da Universidade do Porto

Resumo — O eletrooculograma avalia o movimento ocular, através de diferenças de potencial entre a córnea e a retina. O objetivo deste trabalho foi criar uma plataforma de escrita de mensagens, controlada por movimento ocular, para pessoas impossibilitadas de comunicar. Neste sentido, e tendo apenas ferramentas digitais à nossa disposição, foi desenvolvida uma plataforma de simulação de movimento ocular, em LabVIEW, que recorre a 3 segmentos de sinal EOG - repouso, pico positivo e pico negativo -, obtidos na base de dados Physionet. Para simular o processamento analógico, os sinais recolhidos são processados em Multisim, e o seu resultado é enviado para o LabVIEW, por co-simulação. O LabVIEW comunica também com um executável desenvolvido em linguagem Java, correspondente à interface gráfica que o utilizador irá usar. Esta interface permite ao utilizador escrever mensagens de texto, que podem ser enviadas por email, guardadas em ficheiro de texto, ou lidas em voz alta. Para gerir e planear o projeto, 3 ferramentas foram utilizadas – UML, Pencil e Redmine –, e para o desenvolvimento de um logótipo e vídeo promocional, Adobe Photoshop e Adobe Premiere. Foi possível implementar, de forma simples e robusta, uma plataforma de escrita de mensagens, que poderá ser usada em diversas aplicações mais complexas, nomeadamente na área da robótica.

Palavras-Chave — Eletrooculograma, Movimento Ocular, Simulação de Processamento Analógico, Interface gráfica

I. INTRODUÇÃO

O

s olhos são órgãos sensoriais complexos, responsáveis por converter fotões provenientes da luz visível em sinais eletroquímicos. O olho humano funciona como um dipolo elétrico, pois a retina, localizada na parte posterior, apresenta um potencial negativo em relação à córnea, localizada anteriormente. Numa situação de repouso, o potencial elétrico entre as duas regiões oculares é praticamente nulo e estável. No entanto, aquando do movimento ocular, o globo ocular altera a sua posição o que, consequentemente, irá perturbar o equilíbrio entre os dois polos. Como resultado, gera-se uma diferença de potencial, que pode ser detetada na pele através do uso de elétrodos localizados na periferia do olho: quando, durante o movimento ocular, a córnea se aproxima de um dos elétrodos, a retina aproxima-se do elétrodo oposto, permitindo obter a diferença de potencial entre os dois pontos. É possível também detetar, desta maneira, o piscar de olhos, tanto voluntário como involuntário. O sinal resultante desta aquisição é um eletrooculograma (EOG).

O sinal EOG tem uma amplitude muito reduzida, entre os 50 e 3500 μV e uma gama de frequências entre 0,1 e 100 Hz, sendo que, no entanto, a gama de interesse para análise normalmente situa-se entre os 0,05 e os 30 Hz¹. A recolha deste sinal pode ser relativa a movimentos horizontais ou verticais e, usando seis elétrodos (quatro para medição de potencial e dois de referência) é possível fazer a recolha dos dois sinais

simultaneamente. A recolha do sinal EOG é útil para exames oftalmológicos, mas a sua aquisição e análise também permite o controlo de sistemas.

Neste trabalho, o EOG foi usado com o fim de controlar um sistema de escrita de mensagens, com diversas funcionalidades. Para isso, desenhou-se uma interface Java capaz de reconhecer a direção e o sentido do movimento ocular, permitindo ao seu utilizador escolher entre quatro opções (cada uma associada a cada direção de movimento - esquerda, direita, baixo e cima), e ir selecionando tanto as letras para a escrita da sua mensagem, como opções de exportação para esta.

Devido à impossibilidade de adquirir o sinal diretamente do corpo humano, desenvolveu-se uma interface em LabVIEW que possibilita a simulação do movimento ocular, gerando dois sinais EOG, horizontal e vertical, e que envia as alterações em relação ao repouso para a interface Java.

Para a calendarização e divisão das tarefas entre os membros do grupo, o software Redmine foi utilizado, onde cada semana de trabalho foi descrita, enumerando as tarefas planeadas para cada uma e as horas de trabalho dedicadas a estas. O GitHub foi utilizado para a partilha de ficheiros de código entre todos, de forma eficiente.

II. METODOLOGIA

A. Materiais

Os materiais necessários à implementação - *Bill of Materials* (BoM) -, em laboratório, do circuito analógico primeiramente projetado estão presentes na figura 1 do anexo A. A BoM contabiliza os dois circuitos que seriam montados, um para cada direção do movimento ocular.

Relativamente à lista de componentes utilizados no desenvolvimento do circuito em Multisim, esta está presente na figura 2 do anexo A e contabiliza os três circuitos elaborados para cada segmento do sinal.

B. Circuito Analógico Desenvolvido

Tal como já explicado, sendo que o objetivo inicial do projeto consistia na recolha de sinal EOG e posterior conversão A/D, para uso num sistema de comunicação, foi projetado e implementado um circuito analógico em laboratório, para o processamento analógico do sinal fisiológico. É de notar que ter-se-iam utilizado elétrodos de superfície Ag/AgCl, para a recolha do sinal EOG, amplamente usados na medição de sinais fisiológicos. O circuito foi desenhado tendo por base a literatura existente e resume-se de acordo com o esquema na figura 1².

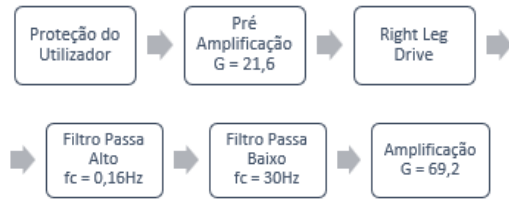


Figura 1 Esquema resumo do processamento analógico

A primeira fase do circuito analógico corresponde ao circuito de proteção do utilizador, que seria fundamental no circuito físico real. Para isso, à entrada do circuito foram adicionadas duas resistências de 100kΩ, impedindo grandes picos de corrente. De seguida, é feita uma pré-amplificação, recorrendo a um op-amp AD620 em modo diferencial, fundamental para reduzir ruído existente. O ganho, sendo dado pela equação 1, foi colocado nos 21,6.

$$G = 1 + \frac{49,4k}{R_G} \quad (\text{Eq. 1})$$

Após a primeira amplificação, o sinal passa por um filtro ativo passa-alto, de 1ª ordem, com um amplificador LM741 e com a frequência de corte dada pela equação 2 e definida em 0,16 Hz, que faz a remoção do offset DC. O filtro apresenta ganho unitário.

$$f_c = \frac{1}{2\pi\sqrt{R}C} \quad (\text{Eq. 2})$$

De seguida, foi implementado um filtro passa-baixo de 2ª ordem, ativo, do tipo *Butterworth*, com implementação *Sallen-Key*, com um amplificador LM741 e com frequência de corte dada pela equação 3 e colocada a 30 Hz, removendo componentes de alta frequência que não são relevantes na análise do sinal EOG, obtendo assim apenas o sinal dentro da gama de interesse. Adicionalmente, remove também ruído de fonte, de 50Hz. O filtro apresenta ganho unitário.

$$f_c = \frac{1}{2\pi\sqrt{R_1 C_1 R_2 C_2}} \quad (\text{Eq. 3})$$

Finalmente, é realizada uma segunda amplificação, garantindo que a gama de voltagem do sinal está entre 0 e 5V, para que o Arduino UNO, que seria utilizado na conversão A/D, o receba corretamente. Para isso, recorre-se a um amplificador LM741 com um ganho dado pela equação 4, colocado a 69,2. Desta forma, a amplificação total esperada é de 1493.

$$G = 1 + \frac{R_f}{R_i} \quad (\text{Eq.4})$$

Seria ainda necessário reforçar o estágio inicial de proteção do utilizador, recorrendo, por exemplo, a condensadores e transístores, que iriam limiar a voltagem e a corrente que atingiria o utilizador, numa situação de curto circuito. Adicionalmente, na junção entre as resistências de R_G , seria implementado o *Right-Leg Driver*, com um LM741 e ganho ajustável entre -1 e -10 com um potenciômetro, para minimizar a interferência de modo comum.

C. Planeamento Global do Projeto em UML

De facto, devido às restrições que se impuseram, foi necessária uma completa reestruturação do projeto, tornando possível realizá-lo com as condições existentes. Assim, ao invés de recolher o sinal fisiológico de um utilizador, foi utilizado um sinal da Physionet, que sofreu um pré-processamento no Matlab, onde se extraíram 3 componentes distintos - sinal em repouso, pico positivo, pico negativo - e que foi exportado em formato csv. Estes 3 segmentos do sinal foram utilizados para simular todo o movimento ocular, recorrendo a LabVIEW, para integrar os diversos segmentos e permitir ao utilizador simular as direções do olho, e a Multisim, para simular o processamento analógico de cada segmento. A interface gráfica do programa foi desenvolvida em Java e comunica, em tempo real, com o LabVIEW, para permitir ao utilizador escolher iterativamente as letras da mensagem.

Para melhor projetar a comunicação entre as ferramentas escolhidas, considerou-se importante desenvolver um diagrama de atividade, em linguagem UML, permitindo um melhor planeamento dos métodos a usar para alcançar todos os requerimentos do projeto.

D. Aquisição do EOG

Pesquisa do Sinal em Bases de Dados

De modo a adquirir o sinal eletrooculográfico, pesquisou-se na base de dados Physionet *sets* de sinais que incluíssem um sinal

EOG. Optou-se por um set relacionado com a aquisição de sinais durante o sono – *Sleep EDF Database Expanded* – que contém vários ficheiros EDF (*European Data Format*), cada um apresentando vários canais com diferentes dados, nomeadamente EEG, EMG, temperatura e respiração e, como pretendido, um EOG horizontal, que foi transferido e utilizado neste trabalho.

Pré-processamento em Matlab

O passo seguinte consistiu na leitura do ficheiro EDF e posterior seleção do sinal EOG. Para isso, utilizou-se o software Matlab e uma extensão deste software que possibilita a leitura deste tipo de ficheiros – *Biosig*. Na figura 2, é possível observar a representação gráfica dos vários sinais presentes nos ficheiros. De seguida, isolou-se o canal do sinal EOG de modo a possibilitar a sua observação e posterior edição; o sinal EOG completo pode ser observado na figura 3A enquanto que na figura 3B está representada a secção inicial do sinal.

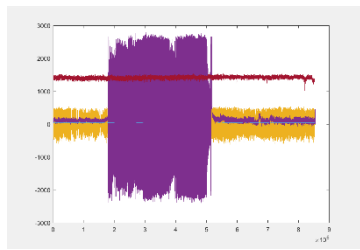


Figura 2 Sinais presentes no ficheiro EDF

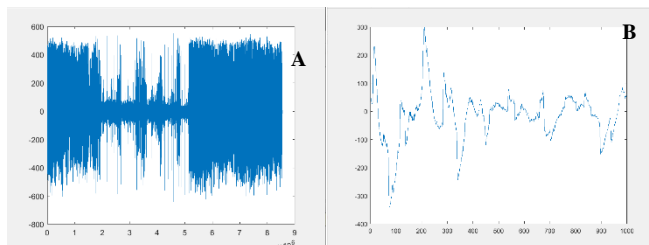


Figura 3 A - Sinal EOG completo, B - Secção inicial do sinal EOG

Finalmente, o sinal foi seccionado de modo a isolar três componentes: um pico positivo, um pico negativo e uma região correspondente ao repouso. Os três componentes são, por fim, convertidos para o formato *csv* e serão posteriormente utilizados para simular o movimento ocular. Uma vez que só foi possível obter um EOG horizontal, os picos e a componente de repouso obtidos serão também utilizados para simular o movimento na direção vertical, já que os formatos são muito semelhantes entre si, não afetando a interpretação do sinal.

Processamento dos componentes em Multisim

Após a obtenção dos segmentos de interesse do sinal, estes serão processados num circuito desenvolvido em Multisim. É de notar que foi necessário implementar algumas alterações no sinal, de modo a possibilitar o seu processamento. Primeiramente, foi necessário redimensionar o sinal e considerar que este estava em mV e não em μ V, uma vez que estes valores eram demasiado pequenos para o programa Multisim funcionar corretamente. Assim, multiplicaram-se os

sinais por um fator de 1000. Obteve-se também uma escala temporal para os sinais, através do período de *sampling* do EOG, correspondente a 0.01s, fornecida pela base de dados. De seguida, os valores de voltagem e tempo resultantes foram lidos a partir de um ficheiro *txt* e usados para gerar um sinal através do bloco *Piecewise Linear Voltage*, em Multisim.

E. Processamento do EOG

Multisim

O passo seguinte de desenvolvimento deste projeto consistiu no processamento do sinal através de um circuito desenvolvido no programa Multisim (versão 14.2) que segue o esquema representado figura 4.



Figura 4 Esquema resumo do processamento do sinal em Multisim

O circuito corresponde a uma recriação dos componentes desenvolvidos no circuito analógico real, mas com valores de ganho ajustados à alteração da escala do sinal.

Assim, tal como no circuito analógico original, fez-se, em primeiro lugar, uma amplificação inicial de menor escala. Para esse fim, utilizou-se o amplificador de instrumentação AD620 e uma resistência, usada para ajustar o ganho. O ganho final desta amplificação é 1.1, e foi calculado através da equação 1.

Seguidamente, o sinal sofre um processo de filtragem de modo a limitar a sua gama de frequências às de interesse. Para isso, projetou-se um filtro passa-alto e um filtro passa-baixo. Utilizou-se o amplificador operacional LM358 em ambos os filtros, diferente do LM741 usado em laboratório, uma vez que este foi identificado como causador de erros de convergência em Multisim.

O filtro passa-alto projetado é ativo, de 1ª ordem e permite remover a componente DC do sinal e *baseline noise*. A remoção da componente DC não representa um problema durante a aquisição do movimento ocular, uma vez que é necessário regressar à posição de repouso (olhar em frente) entre cada movimento, não gerando um offset quando se olha constantemente para a mesma direção. A frequência de corte foi de 0,16 Hz, calculada pela equação 2, e o filtro apresenta ganho unitário.

O filtro passa-baixo é também ativo, mas de 2ª ordem e de configuração *Sallen-Key*. Este filtro tem como função eliminar regiões do sinal sem interesse para análise, mas também ruído de elevada frequência. Este filtro foi projetado para atenuar frequências superiores a 40 Hz. Assim, foram selecionados resistências e condensadores de modo a que a frequência de corte seja 40 Hz, que pode ser calculada através da equação 3.

Relativamente ao processo de filtragem, os diagramas de Bode dos filtros projetados podem ser encontrados nas figuras 1 e 2 do anexo B.

Por fim, faz-se uma segunda amplificação, através de um circuito com duas resistências e feedback negativo; para isso utilizou-se novamente o amplificador LM358, em modo não inversor, sendo o ganho associado a este passo igual a 7 (Eq. 4).

Assim, o ganho total de amplificação teórico é igual a 7,7. O circuito de processamento final encontra-se representado na figura 5.

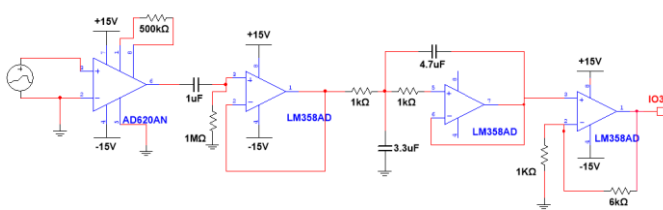


Figura 5 Circuito de processamento do sinal EOG em Multisim

PCB

Depois do desenvolvimento do circuito de processamento de sinal em Multisim, considerou-se uma mais-valia implementar este mesmo circuito numa placa de circuito impresso, ou PCB (*Printed Circuit Board*). O uso de PCBs apresenta inúmeras vantagens em relação à implementação do circuito numa *breadboard*, já que o circuito fica montado de forma definitiva e erros como fios ou componentes mal conectados são evitados. Assim, desenvolveu-se o protótipo do circuito em Ultiboard. Este programa foi escolhido já que apresenta comunicação direta com o Multisim, possibilitando uma transição rápida entre os dois programas. Assim, no circuito Multisim substituíram-se as fontes de tensões e geradoras de sinal por ligações a conectores elétricos *header*, que irão permitir a entrada e saída de sinais no circuito impresso, assim como a sua alimentação. A configuração deste circuito em Multisim foi então transferida para Ultiboard, onde se usaram três camadas de cobre para efetuar as ligações entre os componentes.

F. Simulação do Movimento Ocular

A interface gráfica em LabVIEW (versão 32-bit SP1 2019) é responsável pela simulação do movimento ocular, com input manual do utilizador da direção ocular a resultar na geração de um sinal EOG correspondente. O esquema da figura 6 apresenta, de forma resumida, os principais passos tomados. O diagrama de blocos completo encontra-se no anexo C.



Figura 6 Esquema resumo do diagrama de blocos construído em LabVIEW

O código inicializa-se com a execução automática do ficheiro *jar*, que abrirá a interface gráfica em Java para a escrita de mensagens. A pasta que contém os ficheiros com os sinais produzidos por LabVIEW ou uma mensagem de fim de programa, enviada por Java, é eliminada e novamente criada, limpando ficheiros antigos; tratando-se de uma operação assíncrona, utiliza-se uma *Flat Sequence* para esse efeito. Através de co-simulação com Multisim, cujo diagrama de blocos pode ser observado na figura 7, é adquirido um conjunto de sinais provenientes do circuito analógico simulado em Multisim, que correspondem a um pico positivo, pico negativo ou repouso de EOG, como previamente descrito.

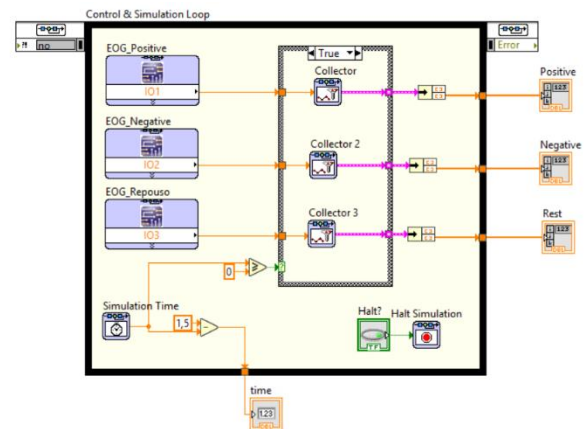


Figura 7 Diagrama de blocos para a co-simulação com Multisim

Com uma estrutura *Control & Simulation Loop*, após um período de inicialização de 5s, durante 1,5s, com um degrau de 0,01s, é recolhido o sinal EOG processado de cada ficheiro Multisim, correspondente a cada estado do sinal. Estes são guardados para serem utilizados ao longo de toda a simulação.

Entra-se, então, num *while loop* que, de acordo com a direção selecionada, irá alocar o sinal EOG ao final da *array* com o sinal total, para cada direção, vertical ou horizontal. Além de ser feito o *display* dos sinais EOG vertical e horizontal, é guardado um ficheiro *txt* sempre que um pico é registado, possibilitando a comunicação com Java, em tempo real. Na figura 8 é possível observar o diagrama de blocos para esta parte.

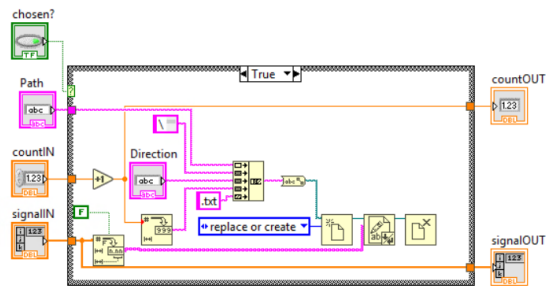


Figura 8 Diagrama de blocos para seleção de EOG

Sempre que um interruptor é ativado, um ficheiro *txt* é criado, contendo no seu nome o número de iteração e a direção – horizontal ou vertical. Os valores do sinal corresponderão a um pico positivo ou negativo, definindo o sentido do movimento ocular, na direção considerada.

Os interruptores estão com retorno de mola, pelo que o movimento lateral só é simulado quando são pressionados ativamente. Assim que o interruptor é libertado, considera-se que o olho se volta a direcionar em frente e o EOG regressa ao estado de repouso. Caso não haja qualquer mudança de direção ocular durante um maior período de tempo, considera-se, igualmente, EOG em repouso.

Finalmente, existe uma sub-rotina de verificação do funcionamento da interface Java. No final de cada iteração do ciclo, caso a pasta dos sinais simulados contenha um ficheiro denominado *EXIT.txt*, gerado pelo Java ao fechar, o LabVIEW para também de correr, de forma síncrona. O diagrama de blocos construído encontra-se na figura 9.

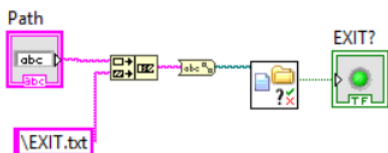


Figura 9 Diagrama de blocos para encerramento síncrono LabVIEW e Java

G. Interface Gráfica

Planeamento em PENCIL

Sendo este projeto desenvolvido para utilizadores com dificuldades motoras, impossibilitados de falar e/ou gesticular, uma das maiores prioridades, no que toca à interface, foi torná-la perceptível e de fácil utilização. Neste sentido, para um melhor planeamento da mesma, realizou-se um primeiro esboço em Pencil, que serviu de ponto de partida para o desenvolvimento da interface em Java.

Desenvolvimento em Java

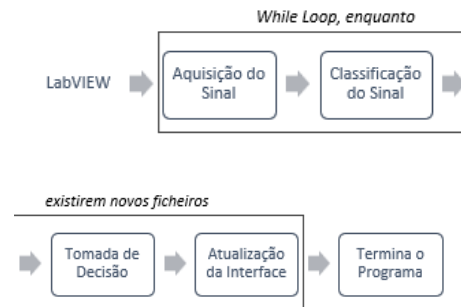


Figura 10 Esquema resumo do algoritmo desenvolvido em Java

Com o intuito de desenvolver uma interface capaz de permitir ao utilizador escrever mensagens através do movimento ocular, diversas fases de processamento têm de ser implementadas. De facto, é necessário desenvolver um algoritmo capaz de receber os ficheiros *txt* criados pelo LabVIEW, classificá-los em relação à direção a que correspondem, e usar essa informação para alterar a interface, que também terá de ser implementada, permitindo ao utilizador escrever a sua mensagem. O esquema presente na figura 10 apresenta os diversos estágios de processamento em Java. É de notar que foi utilizado o programa Eclipse IDE 2020-03 para desenvolver todo o algoritmo de Java.

Após o cuidadoso planeamento da interface em Pencil, desenvolveu-se o código em Java para a implementar. Neste sentido, recorrendo ao plug-in *WindowsBuilder*, criou-se uma *frame* única, com título, logotipo do projeto, um *JTextField*, para mostrar a evolução da mensagem enquanto esta é escrita, e 4 *JTextPanels*, onde são mostradas as opções que o utilizador tem a cada momento.

Ao longo do tempo, as opções serão então atualizadas iterativamente e o utilizador terá oportunidade de escolher, por exemplo, entre uma das 4 letras do alfabeto que serão mostradas na interface. Assim, para melhor organizar as diferentes possibilidades que poderão ser mostradas na interface, desenvolveu-se uma tabela em Excel, posteriormente exportada para formato *txt*, que contém as diferentes possibilidades, organizadas por índices, indicativos das várias direções do movimento ocular que levam o utilizador até cada uma destas. Desta forma, é de salientar que a primeira tarefa desempenhada pelo algoritmo consiste na leitura deste mesmo ficheiro, guardando a informação numa estrutura de dados adequada, que será continuamente usada no decorrer do programa.

Neste sentido, após a classificação da direção de cada sinal, é possível inferir que atualização deverá ser feita no conteúdo dos 4 *JTextPanels* presentes na interface, para que a ação do utilizador seja concretizada.

Relativamente ao primeiro estágio do processamento, a aquisição do sinal, desenvolveu-se um algoritmo que procura novos ficheiros num diretório específico, de 0,1 em 0,1s, e

retorna verdadeiro, caso tenha encontrado um novo ficheiro, ou falso, caso contrário. O tempo total de procura é de 30 segundos ao longo do programa, à exceção da primeira procura, que tem um tempo máximo de 3 minutos, dando tempo ao LabVIEW para realizar a co-simulação.

Se, de facto, for encontrado um novo ficheiro, passa-se ao segundo estágio – classificação do sinal – em que um segundo algoritmo irá ler e classificar o sinal, usando um limiar de valor absoluto 3,5. Assim, se o sinal tiver uma amplitude superior a 3,5, a sua direção será *UP* ou *RIGHT*, consoante a sua origem - circuito horizontal ou vertical. Por oposição, se o sinal tiver uma amplitude inferior a -3,5, a sua direção será *DOWN* ou *LEFT*.

Após ser conhecida a direção do sinal em análise, estão reunidas as condições para passar à terceira fase de processamento – a tomada de decisão, com aplicação do movimento ocular – em que, mediante a direção do sinal, uma das 4 opções mostradas na interface será escolhida e um novo conjunto de opções serão mostradas – atualização da interface –, tal como previamente explicado.

No fim de cada atualização, é procurado um novo ficheiro *txt*: caso este seja encontrado, o ciclo de processamento recomeça, caso contrário, o programa termina. Neste último caso, é criado um ficheiro *txt* denominado *EXIT.txt*, que será posteriormente reconhecido pelo LabVIEW, permitindo que também este pare de correr.

É de salientar que o utilizador, além de conseguir escrever uma mensagem com o programa, tem também diversas possibilidades de exportação à sua disposição. Neste sentido, implementou-se um algoritmo capaz de enviar emails com a mensagem escrita, recorrendo à API *javax.mail*, com um mail da Google criado para o efeito, e um segundo capaz de ler a mensagem final em voz alta, recorrendo à API *com.sun.speech.freetts*. Finalmente, foi também criado um algoritmo que permite guardar a mensagem no computador, num ficheiro *txt*.

H. Dimensão Comercial do Produto

Com o software Adobe Photoshop CC 2019, foi criado um logótipo para o produto em que se materializa o presente projeto. O objetivo foi transmitir uma mensagem de modernidade, simplicidade e robustez, aliando um logótipo apelativo do ponto de vista estético a um slogan inspirador. Da mesma forma, com o Adobe Premiere Pro 2020, foi feita a edição do vídeo promocional do produto, em que se procurou, de forma curta, eficaz e atrativa, mostrar uma possível aplicação do programa desenvolvido.

III. RESULTADOS E DISCUSSÃO

A. Circuito Analógico Desenvolvido

Após a implementação das resistências de proteção, pré-amplificação, filtro passa-alto e filtro passa-baixo, o circuito foi testado com uma onda sinusoidal, com 0,1 V de valor pico a pico (V_{pp}), para uma gama de frequências entre 2 e 100 Hz, para avaliar a eficácia do corte de ambos os filtros, bem como da correta amplificação do final. Os resultados deste teste estão presentes na tabela 1.

Tabela 1 Valores de V_{pp} e ganho para o circuito montado com resistências de proteção, pré-amplificação, filtro passa-alto e filtro passa-baixo, para uma onda sinusoidal de 0,1 V de V_{pp}

Frequência (Hz)	V_{pp} Out (V)	Ganho (V/V)
2	0,600	6,00
10	1,80	18,0
20	1,86	18,6
30	1,50	15,0
40	1,12	11,2
100	0,230	2,23

Tal como esperado, o circuito montado corta as frequências mais baixas do sinal, mais próximas de 0 Hz, e acima de 30 Hz. Assim, obteve-se um sinal com a distribuição das suas frequências na zona de interesse do sinal de EOG, filtrando ruído de fonte, ruído de altas frequências e o *offset* DC. É também possível observar que o ganho máximo do circuito foi de 18,6, obtendo um erro relativo de 13,8%.

Após a verificação do bom funcionamento dos restantes estágios do circuito, foi também testada a segunda amplificação do sinal, também com um sinal sinusoidal, tendo sido obtido um ganho de 72, sendo o erro relativo de 4,1%. Assim, o ganho final obtido foi de 1339, que em contraste com o teórico de 1493, apresenta um erro relativo de 10,3%.

Tal como já mencionado, o *Right Leg Drive* foi projetado, no entanto, não houve oportunidade de o implementar e testar. O circuito final montado está presente anexo D.

B. Aquisição do EOG

Os sinais resultantes da segmentação encontram-se representados nas figuras 11, 12 e 13. Como é possível observar, foi possível isolar um pico positivo e um pico negativo, apresentando estes aproximadamente 450 μV de amplitude absoluta, enquanto a componente relativa ao repouso apresenta uma dimensão menor, apresentando valores entre os -130 μV e os 150 μV .

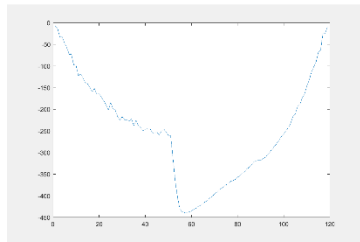


Figura 11 Pico Negativo do sinal EOG

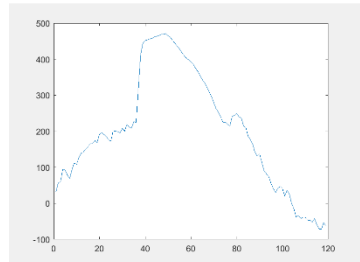


Figura 12 Pico Positivo do sinal EOG

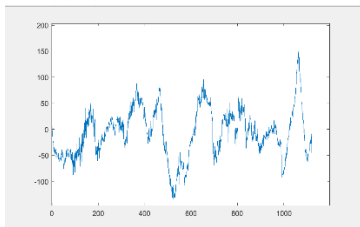


Figura 13 Componente de repouso do sinal EOG

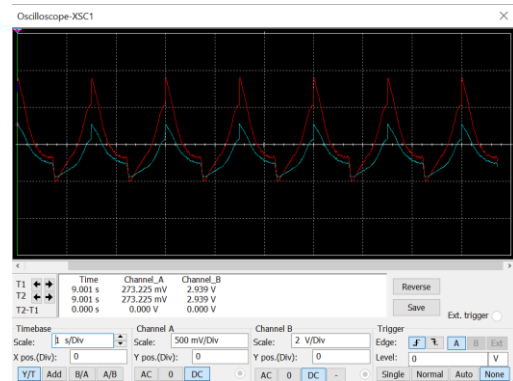


Figura 15 . Pico positivo do sinal EOG visualizado no osciloscópio do Multisim (sinal de entrada a azul e de saída a vermelho)

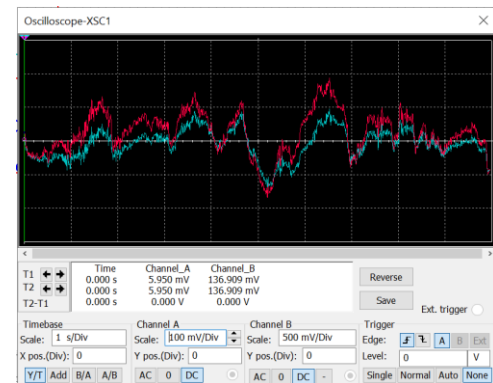


Figura 16 . Repouso do sinal EOG visualizado no osciloscópio do Multisim (sinal de entrada a azul e de saída a vermelho)

C. Processamento do Sinal EOG

Multisim

Após o processamento, obtiveram-se os segmentos de sinal EOG representados nas figuras 14 a 16.

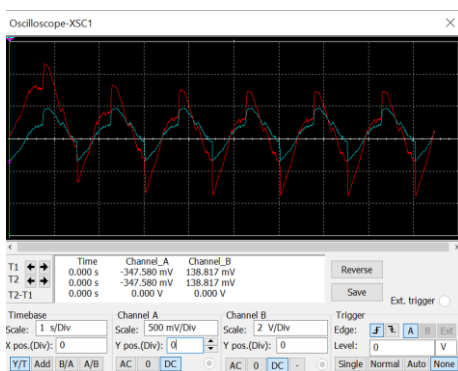


Figura 14 Pico negativo do sinal EOG visualizado no osciloscópio do Multisim (sinal de entrada a azul e de saída a vermelho)

Calcularam-se os valores reais de ganho para cada um dos sinais, comparando os valores máximos de amplitude em pontos correspondentes entre o sinal pré e pós-processado; os valores obtidos encontram-se na tabela 2, juntamente com o erro associado relativamente ao ganho teórico. Como é também possível observar pelas figuras anteriores, os picos do EOG nos sinais de saída, tanto negativos como positivos, apresentam aproximadamente 3,5 V de amplitude.

Tabela 2 Valores de amplitude de entrada e saída dos sinais e ganho real de amplificação de cada componente

	Vin (V)	Vout (V)	Ganho real	Ganho teórico	Erro relativo (%)
Pico Positivo	0,689	5,480	5,966	7,7	3,3
Pico Negativo	0,800	6,421	8,027	7,7	4,3
Repouso	0,211	1,639	7,753	7,7	0,7

PCB

O circuito Multisim alterado para desenvolver o protótipo do circuito impresso encontra-se na figura 1 do anexo E. Neste mesmo anexo também se encontram figuras ilustrativas do protótipo final do PCB: na figura 2 encontram-se representadas as várias ligações nas três camadas de cobre, enquanto que a

figura 3 apresenta uma visualização tridimensional do protótipo final.

D. Planeamento Global do Projeto em UML

Tal como previsto, realizou-se o diagrama de atividades representativo da comunicação entre os diversos softwares utilizados. Neste sentido, foram criadas 4 *swimlanes* - user, LabVIEW, Multisim, Java – que correspondem às entidades envolvidas no decorrer do programa, tornando-se mais perceptível as ações desempenhadas por cada uma e o encadeamento destas no tempo. Foi também criada uma subatividade – *Update Frame according to Signal Direction* – que integra um elevado número de decisões, desempenhadas pelo Java, e que tornaria o diagrama confuso e pouco perceptível, caso estivesse no diagrama principal. O diagrama de atividade encontra-se no anexo F.

E. Simulação do Movimento Ocular

Como se pode ver na figura 17, foi possível criar uma interface em LabVIEW em que o utilizador pode selecionar as direções do movimento ocular e observar o sinal EOG gerado em tempo real. A comunicação com Java funciona de forma igualmente eficaz e de forma praticamente instantânea.

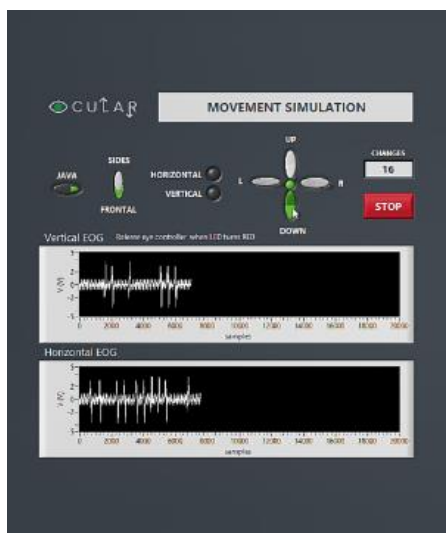


Figura 17 Interface em LabVIEW desenvolvida

Tratando-se de uma comunicação com 3 ficheiros, em simultâneo, o processo de adquirir o sinal coloca o tempo de inicialização de todo o programa nos 2 minutos, que é um período bastante significativo. De seguida, o *while loop* apresenta um *delay* de 2s, ou seja, só é possível mudar de direção de 2 em 2 segundos, valor que foi otimizado para tornar a introdução de nova direção a mais fácil, intuitiva e aproximada à realidade quanto possível.

O resultado final desta secção do trabalho permite, de forma intuitiva e organizada, simular o movimento ocular, de acordo com o objetivo delineado. É importante sublinhar que esta é uma interface mais técnica, com a qual o utilizador incapacitado

não interage, pois serve o propósito de simular o movimento ocular, não recolhido fisicamente.

F. Interface Gráfica

Planeamento PENCIL

Tal como pretendido, a interface a implementar foi primeiramente projetada em Pencil, em que se criou uma frame principal com uma organização simples, tornando o uso do programa mais simples e perceptível. Foi também simulada a escrita de uma mensagem, para melhor perceber como este algoritmo deveria ser implementado em Java. O documento desenvolvido pode ser encontrado no Anexo G.

Desenvolvimento Java

Tal como pretendido, foi desenvolvido um algoritmo, em linguagem Java, capaz de receber os ficheiros *txt* criados pelo LabVIEW, classificar a direção de cada sinal, analisando as amplitudes deste, e atualizar a interface conforme a direção calculada. Na figura 18 é possível observar a *frame* inicial da interface. É de notar que esta interface é adaptada ao ecrã de cada utilizador e, por isso, ao correr o *runnable* do projeto, a janela irá automaticamente abrir com um tamanho correspondente a metade do ecrã do utilizador, para que a outra metade seja ocupada pelo LabVIEW.

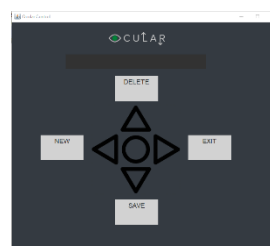


Figura 18 Frame inicial da interface

Adicionalmente, foi também possível implementar as diferentes opções de exportação da mensagem escrita, seja por ficheiro *txt*, que é guardado no computador, por email, a dois contactos distintos, ou por voz, em que o programa lê a mensagem em voz alta. No fim de cada uma destas ações, uma pequena janela sobrepõe-se à *frame* principal, durante alguns segundos, indicando ao utilizador que a ação foi realizada com sucesso. Depois desta fechar, a interface volta à sua disposição inicial. É possível observar na figura 19 um exemplo desta pequena janela, aquando da escolha de enviar uma mensagem por email. Tal como se pode observar, há um pequeno erro na sobreposição da pequena janela, apenas numa zona da interface, para o qual não foi encontrada uma solução.



Figura 19 Interface quando é escolhida a opção de enviar email

G. Dimensão Comercial do Produto

O nome atribuído ao produto criado foi *OCULAR*, transmitindo de forma simples e futurista a capacidade de fazer controlo ocular para escrever mensagens. O slogan criado foi *Beyond Control*, expressando a existência de inúmeras aplicações em aberto para um programa simples, mas robusto, capaz de transformar movimento ocular (simulado) simples em ações mais complexas, através do computador. O logótipo criado foi produzido em duas versões de cor, como se pode visualizar na figura 20, otimizando a sua integração tanto na interface gráfica, como em conteúdos mais comerciais e apelativos. A integração de duas setas, o nome e um olho verde, bem marcado, visa salientar a capacidade de fazer controlo ocular, sem explicitar a palavra *controlo*.



Figura 20 Logótipo desenvolvido, em 2 cores

Em relação ao vídeo, em menos de 1 minuto, foi possível mostrar o funcionamento do programa em computador, uma aplicação real com input manual do movimento ocular e apresentar tanto a imagem como o *slogan* do produto. Na figura 21 podem ser observadas as principais *frames* do vídeo produzido.



Figura 171 3 frames principais do vídeo promocional

H. Análise SWOT

Em jeito de conclusão, foi desenvolvida uma análise SWOT, numa perspetiva crítica sobre o trabalho desenvolvido.

Enquanto ponto forte, identificou-se o facto de se tratar de uma interface simples e intuitiva que possibilita, a um utilizador com limitações de comunicação, escrever mensagens. Além disso, as mensagens podem ser transformadas em voz, enviadas por mail ou guardadas num ficheiro de texto. Com controlo ocular,

pequenas mensagens podem ser escritas num curto período de tempo, com relativa facilidade, após algum treino. No entanto, como aspeto menos forte, e devido às restrições impostas, o movimento ocular tem de ser registado manualmente, não sendo feito de forma automática com elétrodos. Ainda, a interface final depende 3 softwares diferentes, 2 deles especialmente pesados, que não são gratuitos – LabVIEW e Multisim – o que poderá representar uma limitação aquando da transmissão do produto para outro utilizador, nomeadamente numa perspetiva de comercialização.

Diversas oportunidades foram identificadas, entre as quais se destaca a possibilidade de implementar um algoritmo de *Text Prediction* para aumentar a fluidez de escrita, assim como a possibilidade de transformar a mensagem escrita em ações mais complexas, no domínio da robótica, por exemplo. Por outro lado, a aquisição de um sinal EOG real é algo em que este projeto tem bastante espaço para crescer e ser potenciado. No entanto, ameaças são também relevantes, entre as quais se destaca a existência de bastantes produtos semelhantes no mercado dos dispositivos biomédicos, em especial recorrendo a análise de imagem para fazer o seguimento do movimento ocular, o que pode ser mais vantajoso na medida em que erros associados à componente analógica de recolha do sinal são minimizados. Finalmente, movimento ocular excessivo pode ser prejudicial para o utente, podendo causar danos no nervo ótico, o que poderá ser uma limitação caso a comunicação de uma pessoa seja exclusivamente dependente de movimento ocular.

IV. CONCLUSÃO

Neste trabalho, foi criada uma plataforma de comunicação controlada por movimento ocular simulado, intitulada *OCULAR*, que procura servir pessoas com incapacidade para comunicar. Uma mensagem pode ser escrita através de interpretação de sinal de EOG simulado para cada movimento do olho e posteriormente, enviada por email, lida em voz alta, ou guardada em ficheiro de texto.

Diversas ferramentas digitais foram utilizadas: para planeamento e gestão de projeto: Redmine, GitHub UML, Pencil; para obtenção e processamento de sinal: PhysioNet, Matlab, Multisim, Ultiboard; para construção de interface gráfica: LabVIEW, Java; para desenvolvimento da dimensão comercial do produto, Adobe Photoshop e Adobe Premiere.

Assim, três sinais EOG provenientes da PhysioNet, segmentados em Matlab, simulam os três estados possíveis do sinal: pico positivo, pico negativo e repouso. Cada um destes sinais é processado num circuito analógico em Multisim, que por co-simulação é passado ao LabVIEW. Em LabVIEW, através de uma interface gráfica com interruptores para cada direção ocular, o utilizador pode gerar, em tempo real, um sinal EOG pretendido, para qualquer sequência de movimentos. Cada alteração de movimento ocular é guardada num ficheiro

lido pela interface de Java, com a qual o utilizador interage, que permite proceder à escrita da mensagem de texto, com a direções indicadas pelo EOG simulado em LabVIEW.

É também de salientar que seria muito interessante testar o projeto desenvolvido com sinais reais de EOG, recolhidos com elétrodos no utilizador, em tempo real. O trabalho desenvolvido oferece ferramentas para implementar o conceito de controlo ocular para escrita de mensagens de texto, mas implica um input manual do movimento ocular. Se, por um lado, poder ver uma aplicação real, não simulada, do trabalho desenvolvido seria bastante mais recompensador, a adaptação às restrições impostas foi um desafio interessante e encarado com positividade.

Conclui-se que, de facto, é possível interpretar, de forma simples e robusta, um sinal de EOG, transformando-o numa mensagem de texto. Desta forma, abrem-se portas para

aplicações mais complexas, que potenciem o trabalho desenvolvido, munindo doentes com novas e melhores ferramentas para melhorar a sua qualidade de vida, à distância de um movimento do olho.

V. REFERÊNCIAS

- [1] – Iáñez, Eduardo et al. “Using eye movement to control a computer: a design for a lightweight electro-oculogram electrode array and computer interface.” PloS one vol. 8,7 e67099. 3 Jul. 2013, doi:10.1371/journal.pone.0067099
- [2] – López A, Ferrero F, Yangüela D, Álvarez C, Postolache O. Development of a Computer Writing System Based on EOG. Sensors (Basel). 2017;17(7):1505. Published 2017 Jun 26. doi:10.3390/s17071505

VI. ANEXOS

Anexo A – Materiais

Bill of Materials (BoM)					
Item #	Component	Quantity	Manufactor	Description	Destination
1	1kΩ Resistor	4	-	-	Low-pass Filter
2	1,2kΩ Resistor	4	-	-	1ª Amplification
3	100kΩ Resistor	4	-	-	User Safety & Right Leg Drive
4	22kΩ Resistor	2	-	-	2ª Amplification
5	1MΩ Resistor	2	-	-	High-pass Filter
6	1.5MΩ Resistor	2	-	-	2ª Amplification
7	1μF Capacitor	2	-	-	High-pass Filter
8	3.3μF Capacitor	2	-	-	Low-pass Filter
9	4.7μF Capacitor	2	-	-	Low-pass Filter
10	120kΩ Potentiometer	1	-	-	Right Leg Drive
11	LM741 OpAmp	6	Texas Instruments	Industry-Standard Dual Operational Amplifier	Filters & 2ª Amplification
12	AD620 OpAmp	2	Analog Devices Inc	Low Drift, Low Power Instrumentation Amp with Set Gains of 1 to 10000	1ª Amplification
13	ARDUINO UNO REV3	1	Arduino	Microcontroller, Code: A000066	A/D Conversion
14	Breadboard	2	ALL ELETRONICS	Breadboard, 840 contacts, CAT # PB-840	Circuit
15	Eléktrods	5	gvb-spes	Surface adhesive electrode, disposable, snap button Ag/AgCl	EOG Acquisition

Figura A.1 - Lista de Materiais para o circuito analógico inicialmente projetado

Bill of Materials (BoM) - Multisim			
Item #	Component	Quantity	Destination
1	1kΩ Resistor	9	Low-pass Filter and 2nd Amplification
2	6kΩ Resistor	3	2nd Amplification
3	500kΩ Resistor	3	1st Amplification
4	1MΩ Resistor	3	High-pass Filter
5	1μF Capacitor	3	High-pass Filter
6	3.3μF Capacitor	3	Low-pass Filter
7	4.7μF Capacitor	3	Low-pass Filter
8	LM358 OpAmp	9	Filters & 2nd Amplification
9	AD620 OpAmp	3	1st Amplification

Figura A.2 - Lista de Materiais para o circuito desenvolvido em Multisim

Anexo B – Diagramas de Bode para Filtros do Circuito Simulado em Multisim

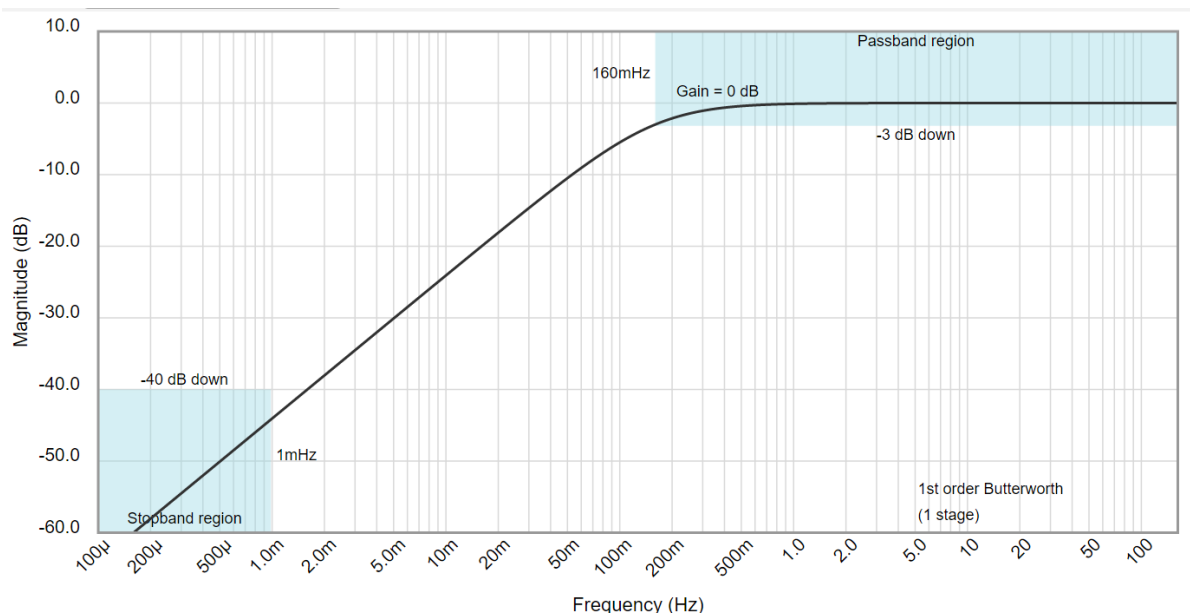


Figura B.1 - Diagrama de bode para um filtro passa alto Butterworth de 1ª ordem com $f_{\text{corte}} = 0.16 \text{ Hz}$

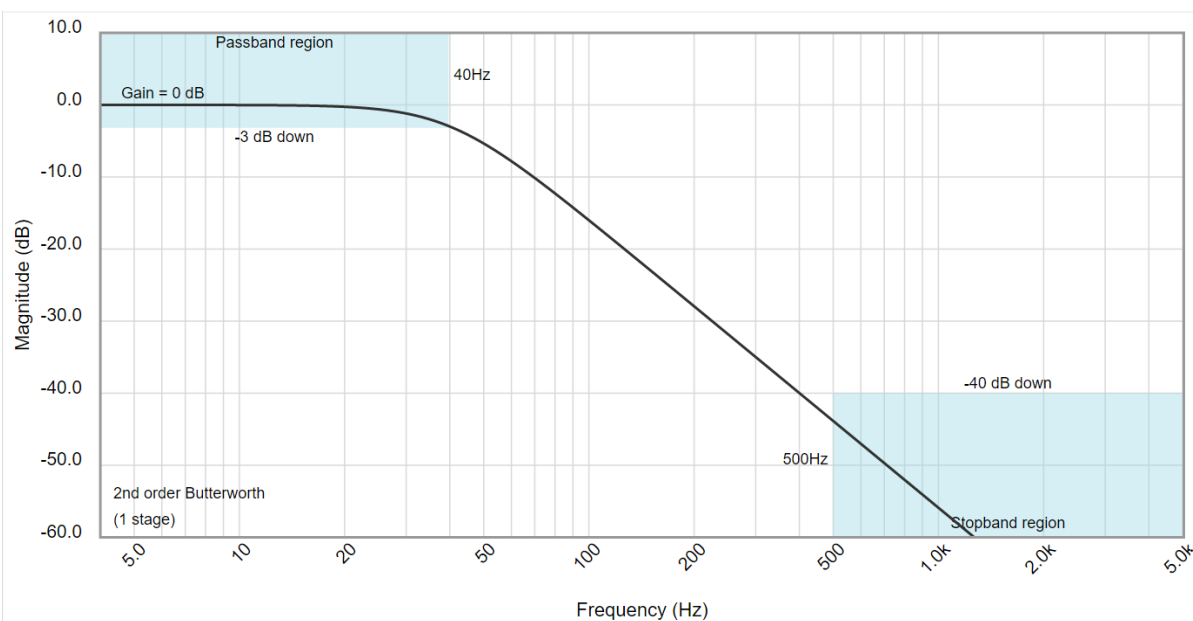


Figura B.2 - Diagrama de bode para um filtro passa baixo Butterworth de 2º ordem com $f_{\text{corte}} = 40 \text{ Hz}$

Anexo C – Diagrama de Blocos e Interface desenvolvidos em LabVIEW

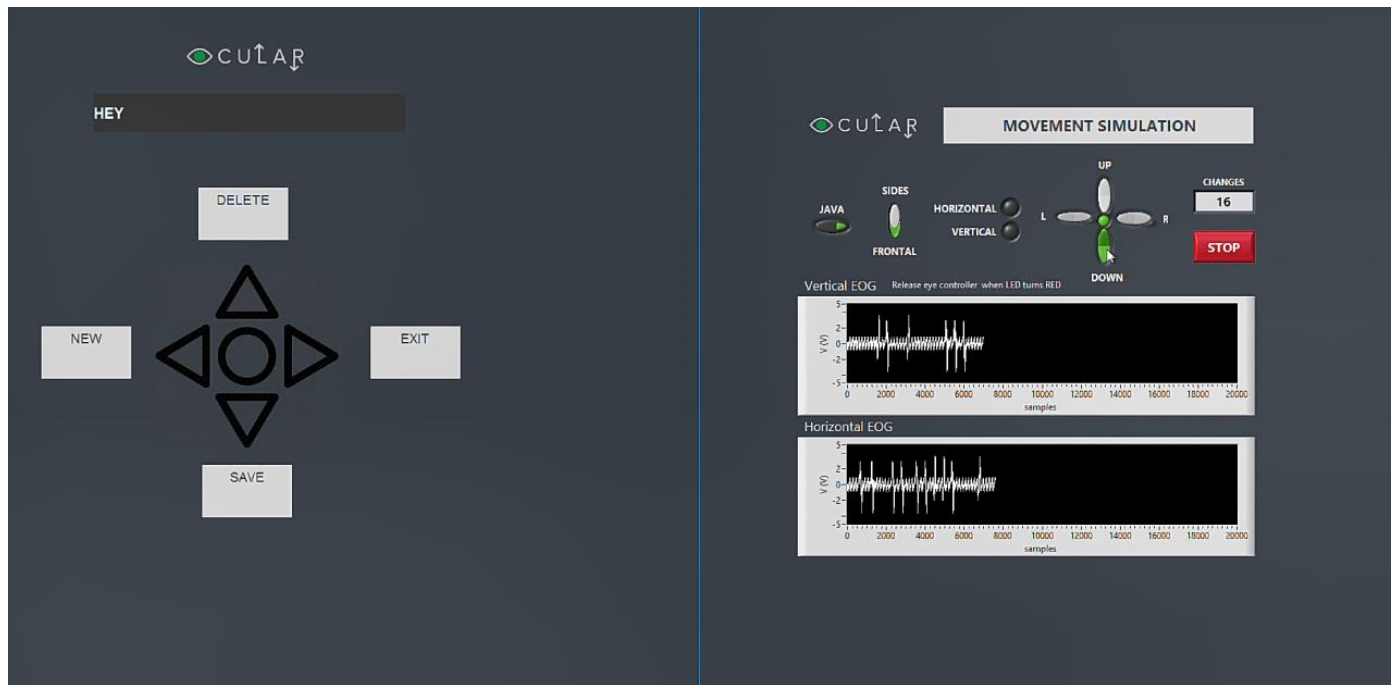


Figura C.1 - Interface gráfica conjunta final, com Java à esquerda e LabVIEW à direita

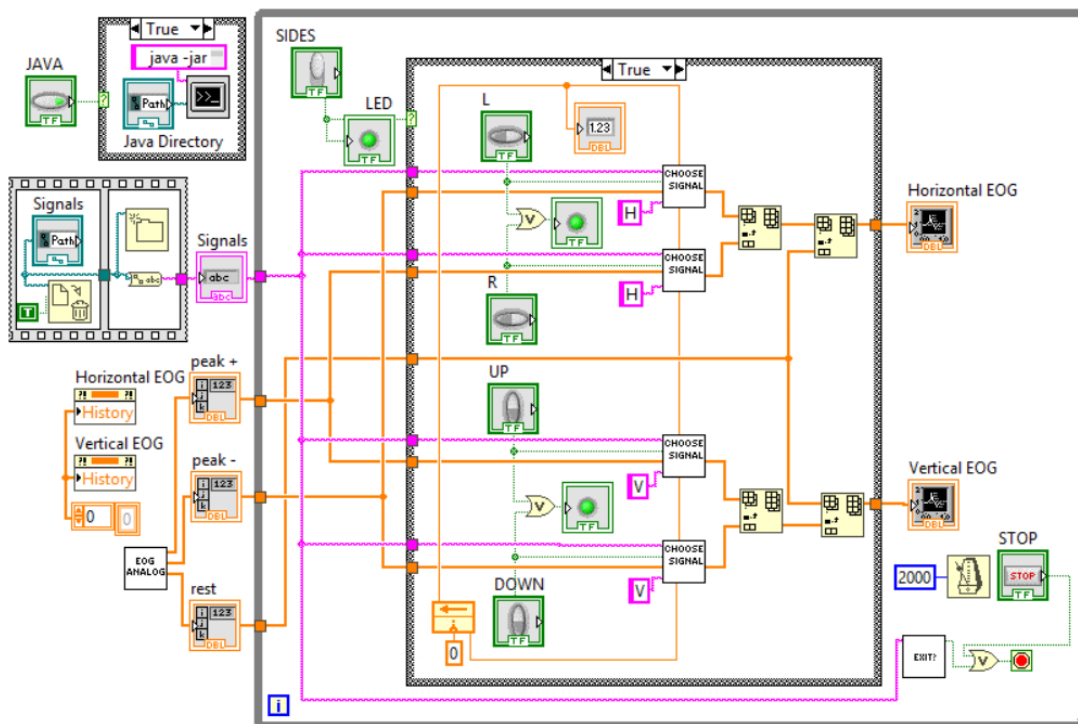


Figura C.2 - Diagrama de blocos desenvolvido em LabVIEW

Anexo D – Circuito Analógico Desenvolvido

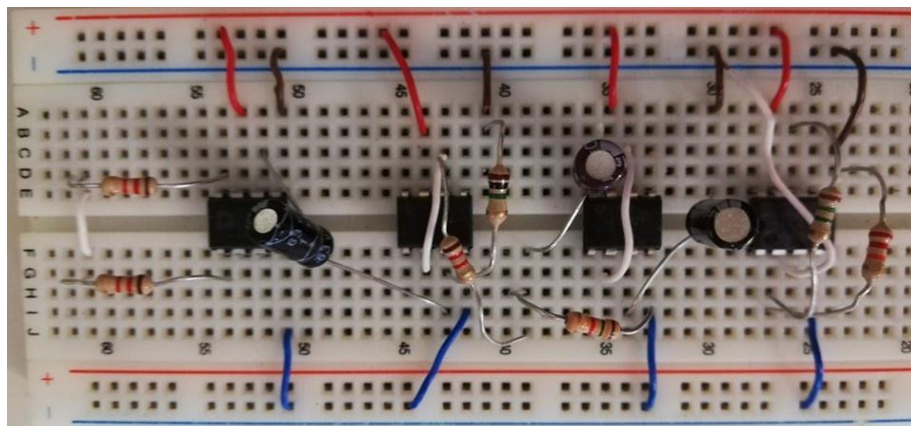


Figura D.1 - Montagem do circuito analógico na breadboard

Anexo E – Circuito Impresso

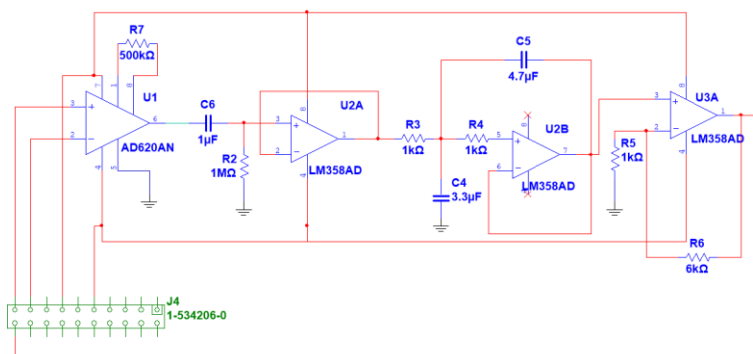


Figura E.1 – Circuito Multisim modificado para o desenvolvimento do PCB

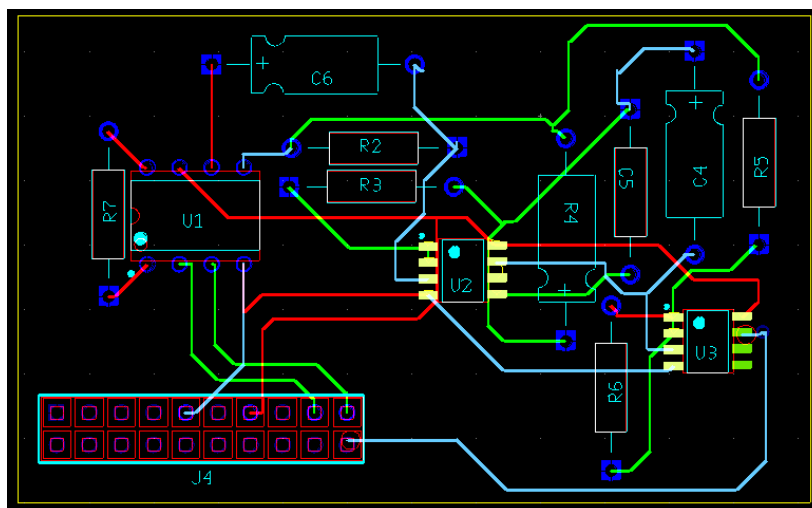


Figura E.2. Protótipo do circuito impresso em Ultiboard

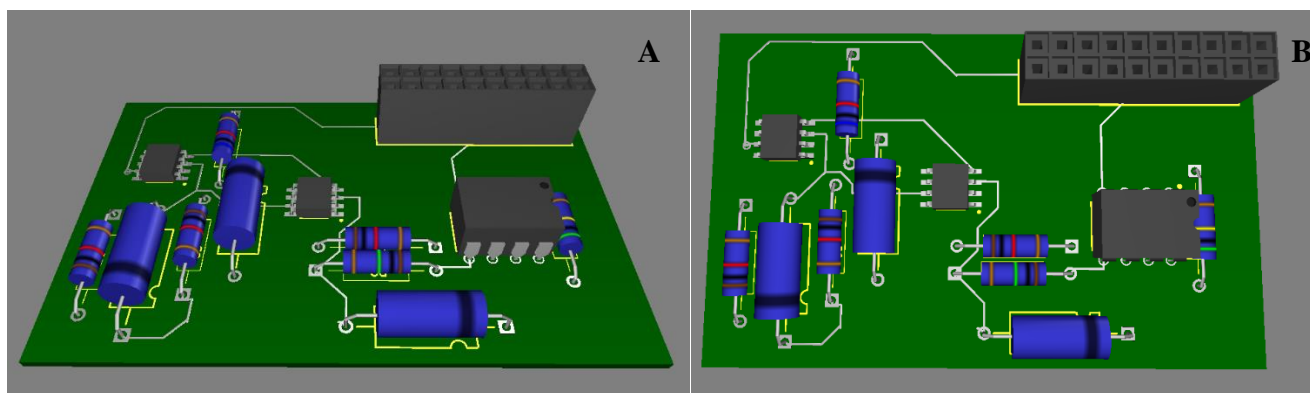


Figura E.3 A,B - Protótipo do circuito impresso – visão tridimensional

Anexo F - Diagrama de Atividade em UML

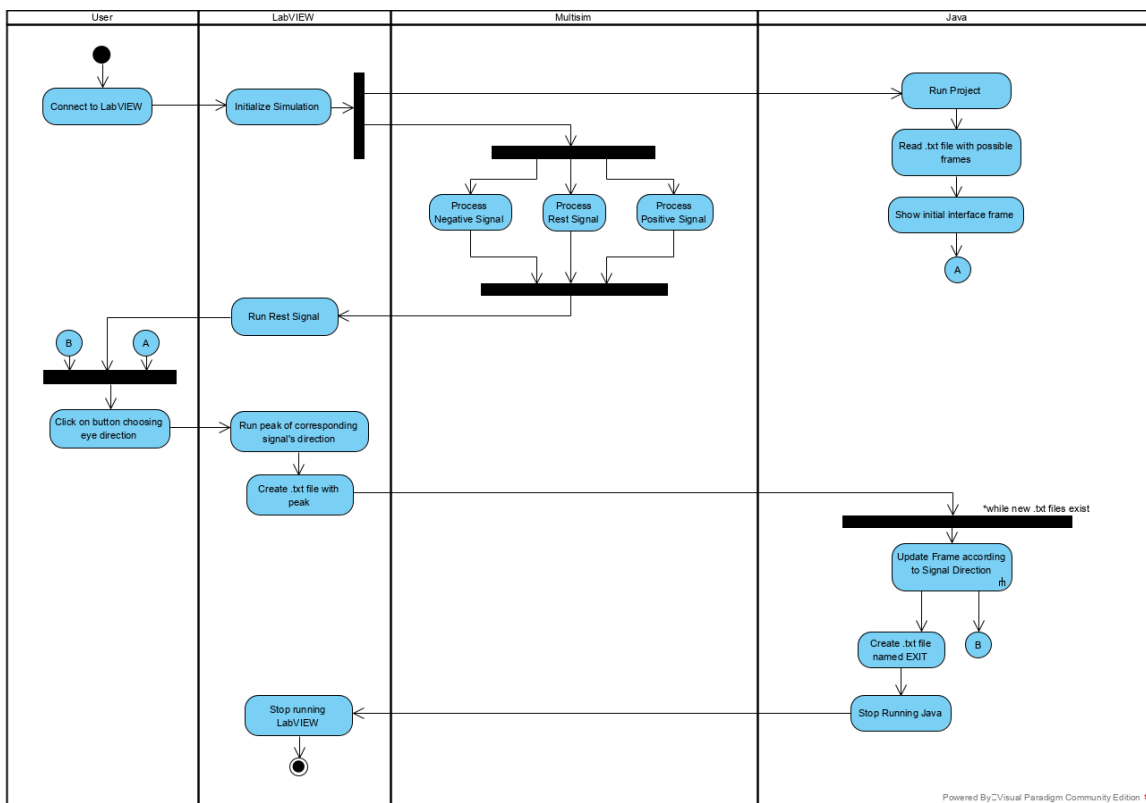


Figura F.1 - Diagrama de Atividade do Projeto

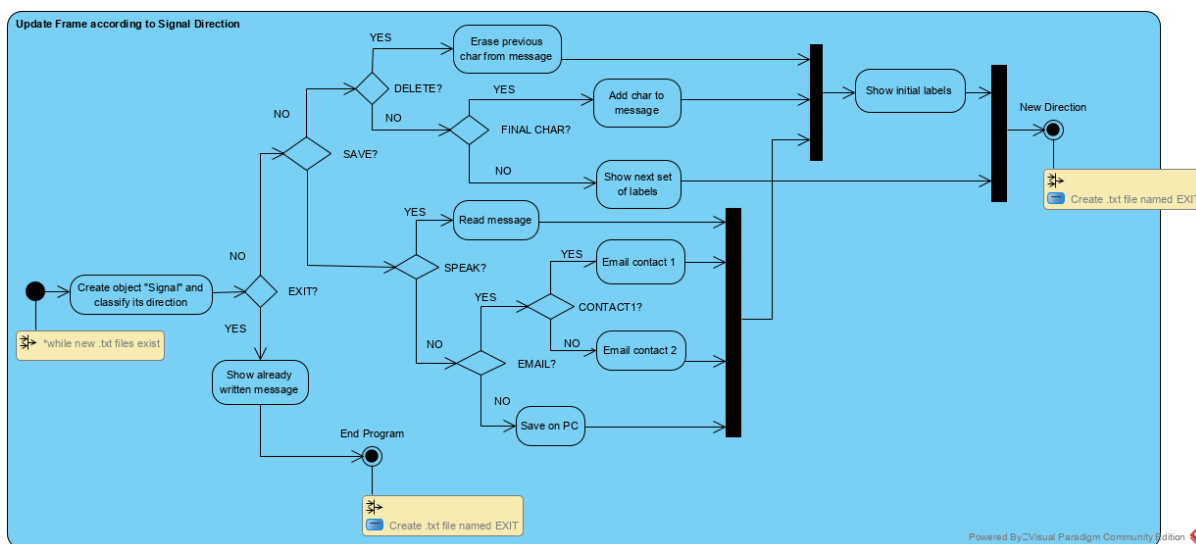


Figura F.2 - Diagrama da sub-atividade Update Frame according to Signal Direction

Anexo G – Interface planeada em Pencil

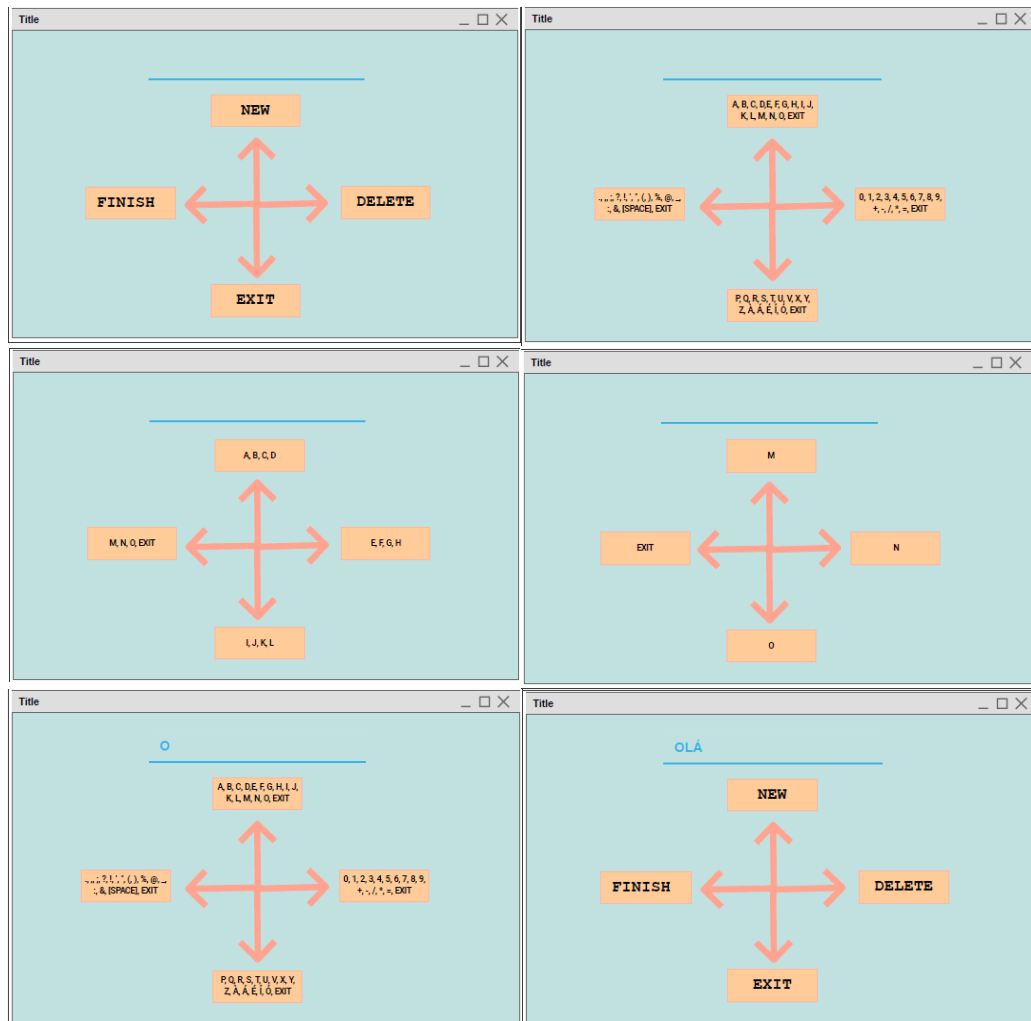


Figura G.1 - Frames de rascunho desenvolvidas em Pencil