

JOÃO HENRIQUE MARQUES VENEZUELA

MARCELO PORTO PERILLO

2050
2
A.M.

Dispositivo de monitoração de exame TD6

São Paulo

2016

JOÃO HENRIQUE MARQUES VENEZUELA

MARCELO PORTO PERILLO

Dispositivo de monitoração de exame TD6

Trabalho de Formatura apresentado à Escola
Politécnica da Universidade de São Paulo
para obtenção do título de Bacharel em
Engenharia

Orientador: Prof. Dr. Henrique Takachi Moriya

São Paulo

2016

Venezuela, João Henrique M.; Perillo, Marcelo P. **Dispositivo de monitoração de exame TD6**. Projeto de Pesquisa (TCC) – Escola Politécnica da Universidade de São Paulo, 2016

Dispositivo de monitoração de exame TD6

Trabalho de Formatura apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do título de Bacharel em Engenharia

Área de Concentração: Engenharia Elétrica com Ênfase em Automação e Controle

Orientador: Prof. Dr. Henrique Takachi Moriya

São Paulo
2016

CATALOGAÇÃO NA PUBLICAÇÃO

AGRADECIMENTOS

Agradecemos a nossos pais e amigos pelo apoio, e ao nosso Professor pela confiança depositada no nosso trabalho e pela ajuda nos momentos de dificuldade.

RESUMO

O desenvolvimento de novos equipamentos e ferramentas é essencial para qualquer campo científico, mas quando se busca um estudo interdisciplinar a tecnologia desenvolvida, mesmo que com pequeno grau de complexidade, pode reverter resultados que beneficiam muito mais os campos envolvidos.

No trabalho aqui apresentado foi proposta a construção de um dispositivo para auxiliar a execução de um exame fisioterápico chamado Teste do Degrau. Nesse teste, um paciente saudável, pneumopata ou cardiopata, executa uma sequência cíclica de subir e descer um degrau, seguindo o ritmo de um sinal sonoro. Tal sinal tem sua periodicidade acrescida dentro de determinados intervalos de tempo, aumentando a dificuldade. Assim, o profissional fisioterapeuta consegue extrair informações acerca da capacidade cardiorrespiratória do paciente com base no tempo em que ele é capaz de executar o teste sem cometer erros.

O degrau desenvolvido possui sensores de contato que deverão auxiliar na detecção daquilo que o fisioterapeuta determina como "erro" na execução do teste. Isso facilitará muito o trabalho do profissional que acompanha o exame, eximindo-o da subjetividade decorrente da observação a olho nu e em tempo real que ele é obrigado a realizar segundo a metodologia atual.

O trabalho aqui desenvolvido é realizado em conjunto com um grupo de alunas de graduação em Fisioterapia do Centro Universitário São Camilo. Trata-se de uma ótima oportunidade para acessar o ponto de vista do profissional que vai trabalhar com o equipamento e também realizar exames com pacientes, a fim de que o equipamento desenvolvido passe por um teste conceitual. É válido lembrar que o equipamento não estará pronto para comercialização, visto que não passou pelos devidos processos de Engenharia de Produto, apesar de seguir as linhas gerais de normas de segurança para o paciente.

Palavras-chave: Teste do Degrau. Paciente. Dispositivo. Interdisciplinar. Arduino.

ABSTRACT

The development of new equipment and tools is essential for any field of science, but when one seeks an interdisciplinary study the technology developed, even with low level of complexity, can provide results that benefit much more the involved fields of knowledge.

The purpose of this work is to build a device to aid the execution of a physiotherapeutic exam called Step Test. On this exam, a healthy patient or a patient with lung disease or heart disease executes a cyclical sequence of stepping up and down on a step, following the rhythm of a artificial sound signal. This signal has its periodicity increased within given time intervals, increasing difficulty. Thus, the physiotherapist can extract information about the cardiorespiratory condition of the patient based on the duration of the exam without mistakes.

The step developed has contact sensors that should help on the detection of the error defined by the physiotherapist. This eases the work of the professional that accompanies the exam, relieving him or her from subjectivity due to naked eye and real time observation that has to be done on the current methodology.

The work developed is realized joining forces with a Physiotherapy graduation group of the Centro Universitário São Camilo. This is an excellent opportunity to access the perspective of the professional that will use the device and also realize exams with real patients in order for the developed device get a concept validation. It's important to emphasis that the device will not be ready for selling, given that it hasn't pass through the normative processes of Product Engineering, even though it complies with standard safety norms.

Keywords: Step test. Patient. Device. Interdisciplinary. Arduino.

LISTA DE ILUSTRAÇÕES E TABELAS

Figura 1 – Figura ilustrativa do Teste do Degrau (TD6).....	13
Figura 2 – Gráfico do sinal de áudio retirado do Matlab.	14
Figura 3 – Degrau montado e aberto, pronto para o teste.	17
Figura 4 – Degrau montado e retraído, pronto para transporte.....	18
Figura 5 – Arduino e sistema elétrico e eletrônico para geração do sinal sonoro e salvamento de relatório do teste.	18
Figura 6 – primeira fase do circuito em protoboard, usado para teste e verificação da programação do microprocessador.	19
Figura 7 – Gráfico do sinal de áudio lido através do Matlab	20
Figura 8 – Gráfico de duração dos ciclos em segundos, em função do número de ciclos	20
Figura 9 – Decaimento do período de cada ciclo ao longo do áudio.....	22
Figura 10 – Imagem do Shield de SD utilizado, compatível com Arduino Uno. Fonte: www.filipeflop.com	22
Figura 11 – Detalhe da parte superior, formada por MDF de 15 mm e 18 mm de espessura.	23
Figura 12 – Protótipo do primeiro degrau projetado no programa SolidWorks.....	23
Figura 13 – Estrutura do segundo protótipo do degrau pronta.....	24
Figura 14 – Esboço esquemático do Pad. Branco: estrutura do degrau. Azul claro: estrutura do Pad com o MDF de 18 mm. Rosa: Pad feito de MDF de 12 mm. Preto: tubos de borracha de 6 mm de diâmetro. Vermelho: placas de alumínio pregadas à madeira. Azul escuro: placas de alumínio para o GND comum, pregadas à parte inferior.....	24
Figura 15 – Detalhe da estrutura superior do degrau inacabada, com os espaços destinados aos Pads.....	25
Figura 16 – Pad inferior encaixado, já com a fiação afixada e devidamente etiquetada.	25
Figura 17 – Detalhe interno dos Pads. Placas de alumínio presas à madeira e fios convergindo para furação passante.....	25
Figura 18 – Detalhe das borrachas laterais. Foram utilizadas apenas nas laterais mais compridas do Pad de modo que o contato não ficasse tão resistente à pressão.....	26
Figura 19 – Detalhe da furação feita na estrutura para proteger os fios que saem de dentro dos Pads.....	26
Figura 20 – Detalhe das placas de alumínio dos contatos pregados e os fios encaixados nos sulcos que levam ao ponto da furação passante presos por grampos metálicos.....	27
Figure 21 – Sinal de áudio desenhado a partir da leitura feita com função do Matlab.	27
Figure 22 – Sinal pós tratamento de modulação, normalização e lógica com treshhold.	28
Figure 23 – Trecho em detalhe do sinal pronto antes de passar pela função de downsampling. Neste sinal a sequência de “uns” não é mais permeada de “zeros”.	29
Figure 24 – Detalhe do Arduino conectado ao circuito da protoboard e ao buzzer. Fase de testes em que ainda não havia sido incorporado o Shield de SD.....	29
Figura 25 – Esquema elétrico Pull-up.....	31
Figura 26 – Circuito de interface entre fios dos Pads e sistema com Arduino. Detalhe para a numeração dos contatos de 1 a 6 e para o sinal de terra comum.....	31

Figure 27 – Protoboard interna conectada aos fios dos Pads. Convencionou-se que os fios vermelhos são advindos dos Pads superiores (1 e 2) e os fios pretos são advindos dos Pads inferiores (3 e 4).	32
Figura 28 – Circuito de teste para debugging da programação em Arduino.	34
Figure 29 – Posições do paciente ao executar o Teste do Degrau. Da esquerda para a direita vê-se a progressão do teste iniciado com o pé direito.	34
 Tabela 1 - Número de ciclos por minuto e seus períodos, teórico e medido	21
Tabela 2 – Tabela de configurações válidas para cada estado da programação. Nas colunas “contatos” temos os sinais digitais esperados para cada um dos Pads: 1, 2, 3 e 4 (pela ordem, pads superiores esquerdo e direito e inferiores esquerdo e direito).....	30

SUMÁRIO

1 DEFINIÇÃO DO PROBLEMA.....	10
1.1 Especificação da necessidade.....	10
1.2 Objetivo	10
2 EMBASAMENTO TEÓRICO	12
3 ESPECIFICAÇÃO DE REQUISITOS.....	15
3.1 Requisitos para o Sistema	15
4 PARCERIA – Centro Universitário São Camilo	16
4.1 Jornada Científica.....	16
4.2 Prêmio Cientista Camiliano.....	16
5 PROJETO	17
5.1 Materiais	17
5.1.1 Degrau	17
5.1.2 Arduino.....	18
5.1.3 Protoboard	19
5.1.4 Matlab	19
5.1.5 Arquivo de áudio	19
5.1.6 Shield SD e cartão micro SD.....	22
6 MÉTODO	23
6.1 Montagem do Degrau	23
6.2 Montagem dos Pads.....	24
6.3 Tratamento do sinal de áudio.....	27
6.4 Programação do Arduino	29
6.5 Montagem do circuito em Protoboard	31
6.5.1 Protoboard interna.....	31
6.5.2 Protoboard externa.....	32
7 VERIFICAÇÃO DO PROJETO	33
7.1 Teste de contato	33
7.2 Testes de Sensibilidade	33
7.3 Testes Gerais	33
7.4 Resultados dos testes.....	34
8 DISCUSSÕES E CONCLUSÃO	35
REFERÊNCIAS.....	36

APÊNCIDE A	37
APÊNCIDE B	40
APÊNDICE C	41

1 DEFINIÇÃO DO PROBLEMA

1.1 Especificação da necessidade

Tendo em vista a metodologia atual e as dificuldades enfrentadas pelos profissionais da fisioterapia, foi proposta a criação de um dispositivo que pudesse automatizar a execução do teste tornando-o mais acurado e confiável.

Na realização do teste através da metodologia convencional, é preciso que dois fisioterapeutas acompanhem o teste. Enquanto um se dedica a verificar o tempo do teste e contar os ciclos executados, o outro deve se atentar para os passos do paciente para verificar se ele não comete algum erro, consequentemente encerrando o teste. Esta demanda de pessoal qualificado para a execução de um teste simples implicou, no decorrer do tempo, o desuso do teste como método de avaliação da capacidade cardiorrespiratória de pacientes saudáveis ou não apesar de sua facilidade em levantar informações sobre o paciente. Além disso, o Teste do Degrau evoluiu para diversas derivações em sua metodologia de aplicação, o que torna os resultados impossíveis de serem comparados entre métodos diferentes.

Neste contexto, há entre os profissionais da área a necessidade de reverter o desuso do Teste do Degrau, visto que é um teste muito simples de ser aplicado e pode ser aplicado até mesmo em lugares remotos. A fim disto, é necessário uma unificação de dados, que em última instância significa uma unificação de metodologia. Para que haja esta padronização do teste e ao mesmo tempo aumento da confiabilidade, o advento de um dispositivo que automatiza o teste pode ser capaz de tornar a aplicação deste exame uma referência no campo da Fisioterapia.

1.2 Objetivo

Este trabalho de conclusão de curso tem por objetivo criar um protótipo de dispositivo degrau que auxilie o profissional fisioterapeuta na execução fidedigna e acurada do Teste do Degrau seguindo as regras do exame, e extraíndo informações através dos contatos instalados no degrau para que o fisioterapeuta possa realizar um diagnóstico mais preciso sobre a condição cardiorrespiratória do paciente.

No decorrer deste projeto, o grupo terá contato com equipes multidisciplinares, expandindo seus conhecimentos e pontos de vista sobre o desenvolvimento de equipamentos eletromédicos. Além disso, a montagem do protótipo permitirá ao grupo enfrentar dificuldades e indeterminações decorrentes do processo de criação e desenvolvimento de equipamentos.

O degrau deverá respeitar as dimensões e normas de segurança para equipamentos aplicados na área da saúde. Além disso, o projeto como um todo será apresentado na Jornada Científica do Centro Universitário São Camilo, concorrendo a um prêmio.

Por fim, o projeto será dado como um sucesso quando a equipe de fisioterapeutas parceira fizer exames comprobatórios da eficácia do novo protótipo em comparação ao exame executado da forma tradicional, o que permitirá um próximo passo para o desenvolvimento do protótipo enquanto produto aplicável em mercado.

A maior finalidade deste projeto será produzir resultados que comprovem a eficiência do novo método de aplicação do teste do degrau, a fim de que o exame aplicado desta forma possa servir como referência para determinar o nicho, e unificar a aplicação do exame. Isso trará maiores bases de dados para comparação, aprimorando não só os exames individuais como também as comparações entre pacientes com condições de saúde semelhantes.

2 EMBASAMENTO TEÓRICO

Em contato com o grupo de fisioterapeutas do Centro Universitário São Camilo, o grupo recebeu embasamento teórico e bibliográfico para determinar as características do projeto, bem como as razões para o seu desenvolvimento.

A avaliação da capacidade física dos pacientes é importante por que reúne aspectos relacionados ao grau de dispneia, trocas gasosas, composição corporal e a tolerância ao exercício. Estes fatores são indicadores notáveis da condição de saúde da pessoa.

O exame realizado para a avaliação da capacidade física pode ser simples ou complexo. Abaixo são citados os testes mais comumente aplicados:

- Teste do degrau de seis minutos (TD6), em que o paciente sobe e desce um degrau de maneira cíclica obedecendo ao ritmo de um estímulo sonoro, enquanto a equipe de fisioterapeutas observa os erros.
- Teste de caminhada de seis minutos (TC6), considerado o mais popular e mais simples, dado que requer equipamento mínimo: apenas um corredor de trinta metros, um temporizador e um oxímetro de pulso.
- Shuttle test, onde o indivíduo irá caminhar entre dois estímulos sonoros que irão ditar o ritmo da caminhada, não há limite de tempo e o teste termina quando o indivíduo não acompanha o ritmo empregado.
- A Ergoespirometria é um teste complexo e mais preciso para avaliar a capacidade física. Os ergômetros convencionais são a esteira e o cicloergômetro.

Os testes são métodos simples e refletem o impacto da doença através do desempenho cardiopulmonar do paciente. O resultado permite estabelecer um programa de reabilitação adequado às limitações e gravidade da doença apresentada pelo indivíduo.

Dentro das variações dos testes físicos funcionais, os testes do degrau têm sido utilizados na avaliação desses pacientes por serem testes práticos e de fácil execução dentro da rotina clínica.

No entanto os testes do degrau utilizados em indivíduos saudáveis e doentes apresentam desvantagens sobre o seu uso, visto que o desempenho pode variar de acordo com o grau de motivação do indivíduo além de serem limitados pelo tempo e apresentar critérios subjetivos para a interrupção dos testes, que é determinada através da observação a olho nu pelo profissional que acompanha o exame. Além disso, existe uma grande diversidade de protocolos que se caracterizam como teste do degrau, dentre eles: o teste de dois degraus de Master, o teste do degrau de Harvard, o teste do degrau de Astrand-Ryhming, testes do degrau gradativo e o teste do degrau da Queen's College. Todos estes métodos usam parâmetros diferentes, como altura do degrau e tempos de aplicação. Até hoje não existem protocolos e tamanhos de degrau padronizados o que contribui para maior dificuldade de avaliação e interpretação dos resultados em tempo real pelo examinador.

A partir de uma reunião inicial com as colegas fisioterapeutas do Centro Universitário São Camilo, o grupo depreendeu o procedimento de execução do Teste do Degrau e pode entender

quais são as dificuldades para o profissional da saúde que aplica o teste. Além disso, o grupo pode fazer medições de um degrau convencional, que serviriam de base para a criação da estrutura do protótipo degrau.

O *Teste do Degrau em seis minutos incremental cadenciado externamente* é aplicado atualmente com o paciente em pé direcionado para uma plataforma, chamada de degrau, que possui 60 cm de largura por 60 cm de comprimento e 20 cm de altura. O paciente deve, então, seguir o comando sonoro dado por um áudio de computador que determina o ritmo da execução do teste. A cada som emitido pelo áudio, o paciente deve mover um membro inferior, primeiro pisando no degrau, depois subindo completamente com as duas pernas. Em seguida, ainda seguindo o ritmo dos estímulos sonoros, o paciente deve descer do degrau, começando pela mesma perna que subiu. Assim que o paciente volta para a posição inicial conta-se um ciclo, e ele deve repetir a sequência enquanto for capaz de acompanhar o áudio.

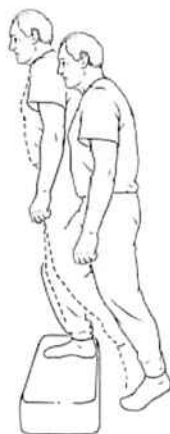


Figura 1 – Figura ilustrativa do Teste do Degrau (TD6).

O áudio do teste emite apitos que indicam quando o paciente deve mover a perna, e a cada intervalo de tempo estes apitos ficam mais rápidos, de modo que a dificuldade do teste se intensifica. Em sua totalidade o áudio chega até 30 minutos, porém é muito difícil acompanhar o ritmo mais acelerado, difícil até mesmo para atletas profissionais que naturalmente possuem um condicionamento físico notável.

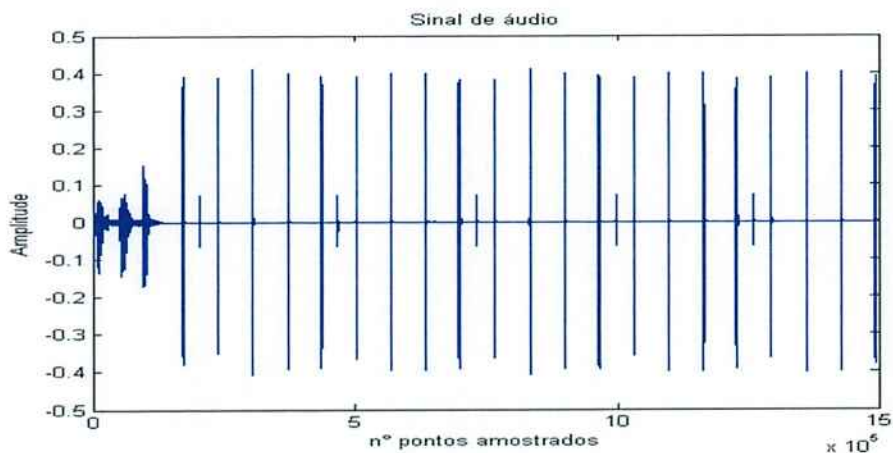


Figura 2 – Gráfico do sinal de áudio retirado do Matlab.

Enquanto o teste está sendo aplicado é necessário que uma equipe de no mínimo dois fisioterapeutas faça o acompanhamento. Ao passo que um deles verifica se o paciente está sincronizando seus passos com o ritmo ditado pelo áudio, o outro deve observar se o paciente não comete erros na sequencia dos passos, bem como se ele não pisa de maneira inapropriada no degrau, com parte da planta do pé para fora do degrau. O teste se encerra quando a equipe detecta um erro, e marca-se o tempo de duração, número de ciclos e qual foi o erro que causou o término.

Abaixo segue listagem do que se caracteriza como erro que necessariamente provoca o fim do teste:

- Assincronia entre passo do paciente e ritmo do áudio. Isto é, se o paciente não obedece ao comando de áudio antes que o próximo comando toque;
- Alteração da sequencia de passos. Ou seja, se o paciente sobe ou desce do degrau com o membro que fez o movimento imediatamente anterior;
- Pisar com parte da planta do pé fora do degrau. Por exemplo, se parte da região calcânea plantar fica para fora do degrau;
- Pisar com parte da planta do pé. O paciente deve sempre pisar com o pé inteiro, dos dedos ao calcanhar, aplicando seu peso integralmente.

3 ESPECIFICAÇÃO DE REQUISITOS

Nesta seção descreve-se a necessidade que o grupo investigou com a equipe de fisioterapeutas do Centro Universitário São Camilo, a fim de atender todas as necessidades e montar um dispositivo que atendesse a todas as normas de segurança.

3.1 Requisitos para o Sistema

Conforme as bibliografias mais consagradas sobre o Teste do Degrau, o ponto mais crucial dos requisitos está relacionado ao dimensional do degrau. O valor mais usual para a altura do degrau é de 20 cm, cota dada como mais eficiente, pois submete o paciente a esforço submáximo, ideal para verificação de condicionamento físico. Além disso, é necessário que haja uma superfície ampla para que o paciente execute o teste, então, seguindo os dimensionais do degrau convencional disponível com a equipe parceira, a superfície do degrau possui 60 cm de comprimento por 60 cm de largura, o que dá uma área mais do que necessária para um pé humano muito maior do que a média.

Como norma de segurança, é necessário que o degrau possua uma superfície antiderrapante onde o paciente pisará, de modo que se reduzam as chances de queda acidental por escorregamento. E como o dispositivo é móvel, foi solicitado também que a base do degrau fosse emborrachada, para que durante o teste, a movimentação do paciente não deslocasse a estrutura do degrau, provocando possíveis acidentes e invalidação do teste.

4 PARCERIA – Centro Universitário São Camilo

Interdisciplinaridade é um dos grandes destaques deste projeto, visto que o trabalho desenvolvido, apesar de não ter grande aprofundamento teórico, causa um impacto direto sobre a aplicação de um exame importante para a aferição das condições físicas de pacientes cardio ou pneumopatas.

Através do nosso professor orientador, tivemos contato com colegas fisioterapeutas e alunas de graduação que decidiram tocar um projeto de conclusão de curso paralelo a este, a fim de se beneficiar da montagem de um equipamento que pode ter impactos profundos na maneira como o teste do degrau é realizado e visto atualmente.

Para este projeto de cunho interdisciplinar, a coisa mais importante foi descobrir com a outra parte, quais são os aspectos importantes para o TD6, e quais informações a equipe de fisioterapeutas precisaria para traçar o diagnóstico do condicionamento físico do paciente. Informações como tempo de duração do teste, número de ciclos, tipos de erro detectados, e quantidade de erros são números importantes não só para diagnóstico mas também para comprovar a eficácia do protótipo.

4.1 Jornada Científica

Houve um evento no Centro Universitário São Camilo, no qual a ideia do protótipo foi apresentada pelas nossas colegas de graduação em Fisioterapia, e para além disso, o projeto foi cadastrado na Plataforma Brasil. A Plataforma Brasil é uma base nacional e unificada de dados que compila e regulamenta pesquisas feitas com seres humanos.

4.2 Prêmio Cientista Camiliano

Durante a Jornada Científica, nossas colegas concorreram a um prêmio para o projeto desenvolvido. E é de maneira extremamente gratificante que o grupo foi vencedor do Prêmio Cientista Camiliano, reconhecendo que a ideia dos dois projetos em paralelo é muito importante do ponto de vista de interdisciplinaridade, mas também é um projeto que pode recuperar a aplicabilidade do TD6.

5 PROJETO

Não só o degrau, mas também os circuitos em protoboard e os programas em Matlab e Arduino foram se modificando no decorrer do projeto. Nesta seção é descrito tanto os materiais utilizados para a montagem do projeto integral, bem como o processo de montagem do protótipo final.

5.1 Materiais

Estudando o degrau convencional visto, o grupo coletou informações de medida para que o degrau pudesse ser reproduzido com fidelidade, incluindo o projeto de contatos para aquisição das informações.

Além disso, foram desenvolvidas diversas ideias em um brainstorm para levantar qual seria a melhor solução para aquisição dos dados durante o teste. Várias ideias foram descartadas e sempre foi escolhida a solução mais simples e que permitisse resultados satisfatórios.

A seguir há uma descrição dos materiais utilizados para a fabricação do hardware (o protótipo do degrau) bem como os equipamentos eletrônicos para a aquisição e tratamento dos sinais que entregariam os resultados. A descrição e detalhes da montagem em si ficam para a seção seguinte, em Método.

5.1.1 Degrau

O cerne do projeto é um protótipo feito em madeira, contendo contatos elétricos sob placas também de madeira. Desses contatos, saem fios que levam a informação da pisada do paciente através de circuitos analógicos até um microprocessador Arduino Uno. O degrau possui uma superfície quadrada de 60 cm e cota de 20 cm entre as plataformas inferior e superior. Além disso, o dispositivo conta com uma cobertura de borracha em ambas as partes.

As figuras 3 e 4 mostram o degrau nas suas duas formas possíveis: aberto e fechado. Esta característica foi incorporada no projeto para atender a necessidade de o degrau ser transportável, apesar de que dada a quantidade de material utilizado, o protótipo tem um peso excessivo para ser carregado por apenas uma pessoa.



Figura 3 – Degrau montado e aberto, pronto para o teste.



Figura 4 – Degrau montado e retraído, pronto para transporte.

5.1.2 Arduino

Foi utilizado um Arduino Uno como interface, além de usá-lo para a reprodução do sinal de áudio e aquisição dos sinais dos pads. Isso trará muitas vantagens e facilidades para o projeto, como por exemplo, possibilidade de tratamento dos sinais analógicos, fácil programação do dispositivo e facilidade em transportar os dados do Arduino para programas como o LabVIEW dada a imensa quantidade de conhecimento disponível para ambos. Vale notar também que o Arduino possui uma ampla base de projetos disponíveis, assim como é fácil encontrar muitos circuitos impressos compatíveis com a plataforma. Desses circuitos, ou Shields, será usado um módulo para leitura de cartão de memória SD conectado ao Arduino para armazenamento do arquivo de áudio tratado, e também para gravação do relatório do exame.

Na figura 5 mostra-se o Arduino integrado ao sistema de circuito em protoboard, Shield SD e buzzer.

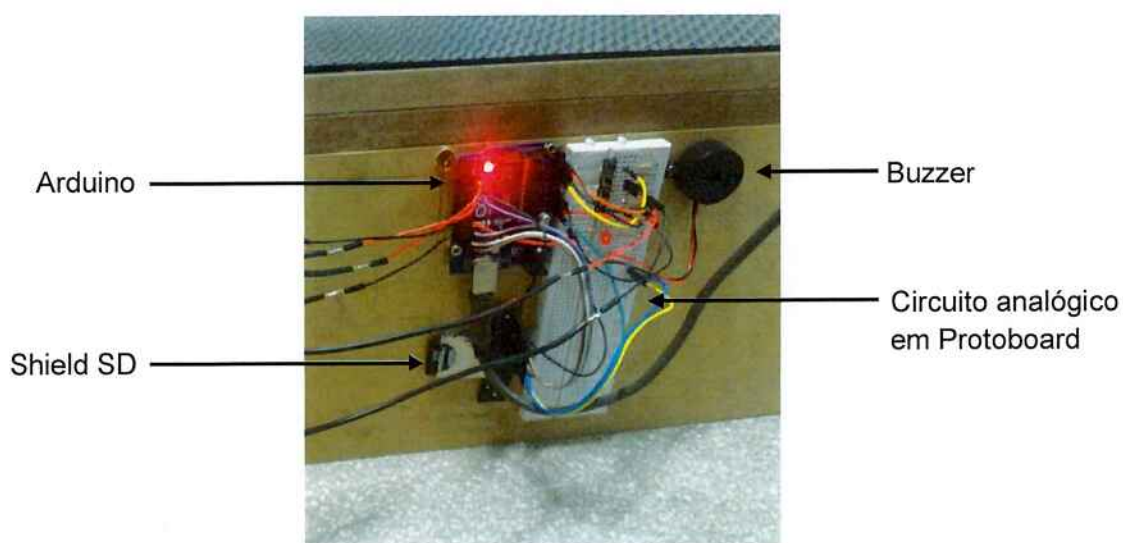


Figura 5 – Arduino e sistema elétrico e eletrônico para geração do sinal sonoro e salvamento de relatório do teste.

5.1.3 Protoboard

Para integrar o Arduino com a montagem física foi realizada a montagem de um circuito em duas protoboards. Este circuito também foi usado com ferramenta de debugging, tanto da programação do arduino quanto da montagem física. A primeira versão do circuito em protoboard é mostrada na figura 6. Essa montagem consistia em um conjunto de 4 botões em paralelo com os contatos dos Pads e 4 LEDs. Tanto os contatos/botões e os LEDs foram conectados ao Arduino.

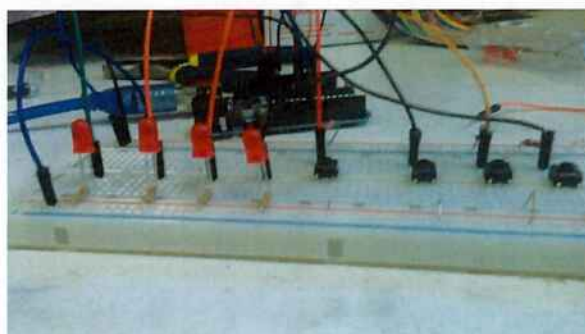


Figura 6 – primeira fase do circuito em protoboard, usado para teste e verificação da programação do microprocessador.

5.1.4 Matlab

O sinal sonoro utilizado normalmente é um arquivo .wav tocado em geral por um computador. Para que o grupo possa fazer a detecção de erros é necessário que este sinal seja integrado ao Arduino para que o microcontrolador possa, além de transmitir o sinal, identificar se os sinais dos contatos estão condizentes com o estímulo sonoro.

Para isso, foi necessário usar o programa Matlab para traduzir o áudio .wav para um arquivo de texto (.txt) de sinal binário (uns e zeros) que pudesse ser salvo no Arduino e usado para os fins necessários.

A implementação do programa em Matlab precisou de modulação, e lógica para efetuar downsampling no arquivo original porque era necessário que o arquivo de áudio se tornasse uma sequência de “uns” e “zeros” para que o Arduino compreendesse o arquivo como digital.

5.1.5 Arquivo de áudio

O áudio utilizado é um arquivo ‘.wav’ de duração de trinta minutos e quatro segundos, que produz estímulos sonoros similares a sibilos de curta duração, que devem comandar o passo do paciente durante o exame. Os toques seguem um padrão cíclico de um toque mais longo, seguido de três toques curtos, definindo um ciclo completo do exame, pois totalizam-se quatro movimentos, o que é necessário para o paciente sair da posição inicial e retornar a ela.

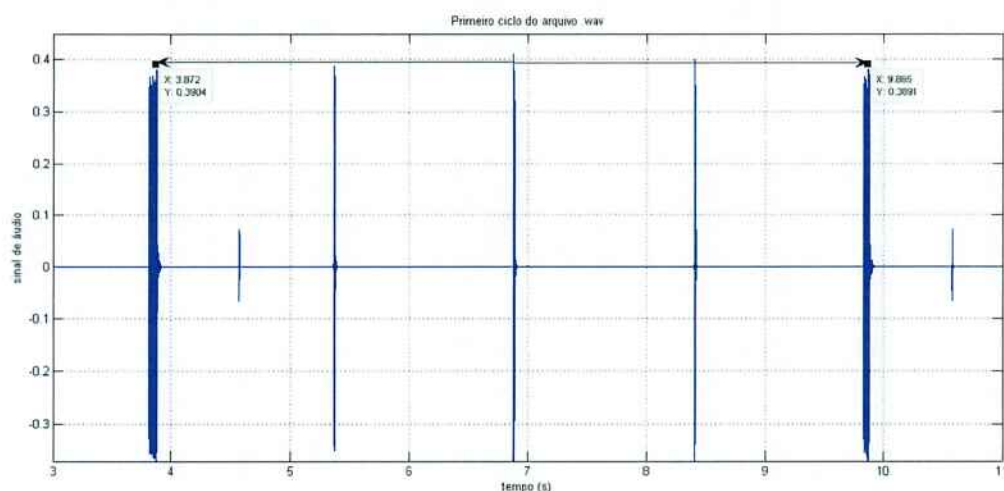


Figura 7 – Gráfico do sinal de áudio lido através do Matlab

À medida que o áudio progride, a periodicidade do ciclo aumenta, tornando os estímulos sonoros mais acelerados, aumentando o grau de dificuldade do exame. Porém, este incremento de velocidade se dá por estágios. O gráfico da figura 8 mostra os primeiros 6'30" do áudio em que se pode ver nitidamente que o tempo dos ciclos decai e o número de ciclos por patamar se prolonga conforme o transcorrer do áudio.

Foi feito um tratamento do sinal através do Matlab (ver código no Apêndice A), para que o áudio pudesse ser completamente caracterizado, e para fins de confirmação, o áudio foi reproduzido algumas vezes em determinados trechos e o período dos ciclos foi cronometrado. No gráfico da figura 8, vemos em azul os pontos correspondentes à aquisição empírica de valores para os ciclos do início do áudio, e em vermelho podemos ver a curva correspondente aos valores calculados pelo Matlab, confirmando o padrão de mudança na frequência das ocorrências dos sinais sonoros.



Figura 8 – Gráfico de duração dos ciclos em segundos, em função do número de ciclos

A cada minuto, a periodicidade dos ciclos aumenta, de forma que se concluem mais ciclos dentro de um mesmo estágio. A rotina criada no Matlab pode identificar o período de cada ciclo e com estes dados uma média foi calculada para cada trecho do áudio dividido em minutos. Esta média mostrou que o áudio começa com 10 ciclos por minuto, e a cada estágio de um minuto são adicionados dois ciclos, aumentando a velocidade a fim de que o exame se torne mais difícil. Este padrão segue até o 15º minuto, e a partir daí o número de ciclos se estabiliza em 38 ciclos por minuto, conforme se vê na tabela abaixo. Contando empiricamente o número de ciclos completados a cada minuto de áudio foi possível confirmar o cálculo feito através da rotina de Matlab. Na tabela representam-se os 16 primeiros minutos do áudio e mostra-se a comparação entre o valor dos períodos calculados pelo Matlab em relação ao valor teórico seguindo o padrão descoberto. Após o quinquagésimo minuto, o valor do período do ciclo se mantém em aproximadamente 1,57 s até o final do arquivo de áudio.

Tabela 1 - Número de ciclos por minuto e seus períodos, teórico e medido

Minuto	Ciclos por minuto	Período teórico de um ciclo (s)	Período calculado pelo Matlab (s)
1'	10	6,00000000	6,00000454
2'	12	5,00000000	5,00000756
3'	14	4,28571429	4,28571914
4'	16	3,75000000	3,75000283
5'	18	3,33333333	3,33333585
6'	20	3,00000000	3,00000113
7'	22	2,72727273	2,72727479
8'	24	2,50000000	2,50000189
9'	26	2,30769231	2,30769318
10'	28	2,14285714	2,14285795
11'	30	2,00000000	2,00000151
12'	32	1,87500000	1,87500142
13'	34	1,76470588	1,76470722
14'	36	1,66666667	1,66666730
15'	38	1,57894737	1,57894737
16'	38	1,57894737	1,58080612

Estes dados permitem criar um gráfico do decaimento do período conforme transcorrem os patamares, alterados de minuto a minuto.

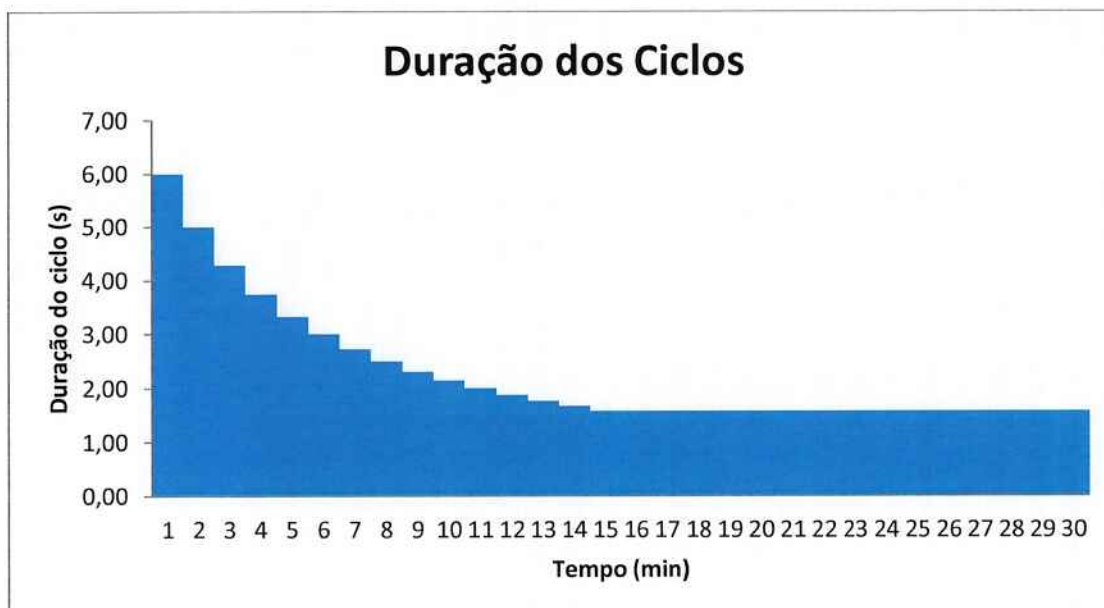


Figura 9 – Decaimento do período de cada ciclo ao longo do áudio.

O arquivo de áudio atinge um nível máximo de 38 ciclos por minuto, isto é, 152 passos por minuto. Para se ter uma ideia, são mais de dois passos por segundo. Para um paciente saudável esta atividade é intensa e submete qualquer indivíduo adulto a grande esforço, por isso não há sentido em intensificar ainda mais a velocidade do exame. Além disso, pacientes com doenças cardiorrespiratórias tem muita dificuldade de atingir esta taxa, e geralmente finalizam o teste antes de chegar a tal nível de esforço. Dificilmente, qualquer que seja o paciente, é raro que o teste passe dos 6 minutos, pois erros de assincronia passam a ser muito comuns.

5.1.6 Shield SD e cartão micro SD

O arquivo .wav com o áudio para o teste possui aproximadamente 30 minutos de duração no seu total. Isso gerou um arquivo proveniente da leitura do '.wav' no Matlab com um número de pontos exorbitante, aproximadamente 79,5 milhões de pontos, e por conta disso foi necessário o downsampling. Mas mesmo assim, com a memória limitada do Arduino, foi necessário adquirir um Shield de cartão SD para que fosse possível ampliar a memória e comportar o tamanho do arquivo com o áudio sem prejudicar a capacidade do microprocessador. Além disso, o cartão de memória micro SD foi utilizado para armazenamento do relatório resultante do exame com o paciente.



Figura 10 – Imagem do Shield de SD utilizado, compatível com Arduino Uno. Fonte: www.filipeflop.com

6 MÉTODO

6.1 Montagem do Degrau

Tal qual o degrau convencional utilizado normalmente no teste do degrau, o protótipo foi fabricado mantendo as dimensões úteis de 60 cm de comprimento, 60 cm de largura e cota de 20 cm entre plataforma inferior e superior. Essas medidas são importantes por que se julga que 20 cm de altura seja um valor ideal para que o teste seja executado sem que o paciente atinja nível de esforço máximo, além do mais, uma área de 60 cm² é razoável para que se faça o exercício sem a distração de ter que mirar onde se pisa para não cometer erros.

A estrutura do degrau desenvolvido é separada em duas partes, a plataforma superior e inferior. A plataforma superior foi feita com MDF de 15 mm de espessura para as laterais e topo. Por cima desta base instalou-se uma placa de MDF de 18 mm, cuja função é garantir resistência a cargas pesadas durante o uso. Sobre esta placa, ripas de MDF foram parafusadas para criar dois espaços, batizados de “*Pads*”, para que fossem instaladas placas também de MDF com os contatos. Já a plataforma inferior foi feita exatamente igual a estrutura da placa de 18 mm da parte superior, também com ripas que formam o espaço destinado aos *Pads* inferiores.



Figura 11 – Detalhe da parte superior, formada por MDF de 15 mm e 18 mm de espessura.

Na figura 12 mostra-se em detalhe a estrutura do primeiro protótipo do degrau conforme descrito acima, criada no programa SolidWorks para visualização do projeto.

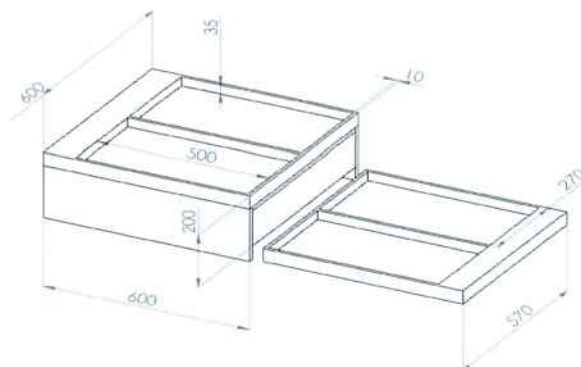


Figura 12 – Protótipo do primeiro degrau projetado no programa SolidWorks.

Abaixo, mostra-se a estrutura do segundo protótipo do degrau. Diferente do primeiro, este já conta com a inversão da plataforma inferior, feita para que os pads inferiores ficassem imediatamente próximos ao desnível do degrau. Esta alteração permitiu que, na parte inferior da estrutura, quatro cantoneiras em “L” fossem parafusadas para que a parte inferior do degrau se afixasse à parte superior. Desse modo, durante o teste a movimentação do paciente não provocará deslocamento indesejado do degrau, evitando problemas para o teste e insegurança para o paciente.



Figura 13 – Estrutura do segundo protótipo do degrau pronta.

6.2 Montagem dos Pads

Os Pads são as plataformas sobre as quais o paciente deve pisar. Essa plataforma se constitui de uma placa de MDF de 12 mm de espessura sob a qual se instalou contatos feitos com placas de alumínio de espessura de 0,3 mm devidamente lixadas. As placas foram conectadas com fios de cobre encapados e afixados através de sulcos feitos na própria madeira e presas com grampos metálicos.

Abaixo há um esboço esquemático de um corte dos contatos que ilustra seu funcionamento. A legenda explica as partes do esboço através das cores.

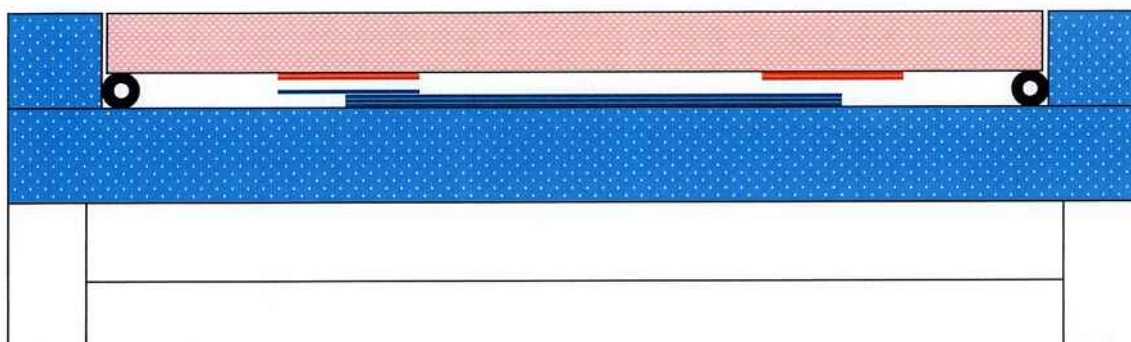


Figura 14 – Esboço esquemático do Pad. Branco: estrutura do degrau. Azul claro: estrutura do Pad com o MDF de 18 mm. Rosa: Pad feito de MDF de 12 mm. Preto: tubos de borracha de 6 mm de diâmetro. Vermelho: placas de alumínio pregadas à madeira. Azul escuro: placas de alumínio para o GND comum, pregadas à parte inferior.



Figura 15 – Detalhe da estrutura superior do degrau inacabada, com os espaços destinados aos Pads.



Figura 16 – Pad inferior encaixado, já com a fiação afixada e devidamente etiquetada.

Para todos os quatro Pads, dois superiores e dois inferiores, há três placas de alumínio inferior com função de terra comum (GND). Estas placas foram pregadas à madeira para afixação e escolheu-se usar três placas em conjunto para que se garantisse o contato. Já na parte superior de cada Pad de contato há seis placas de alumínio menores, dispostas matricialmente (3x2) ajustadas e alinhadas para que todas tenham contato com a placa grande de baixo. Por motivos de ajuste de sensibilidade, foi necessário pregar placas menores tanto à placa grande de GND e quanto às placas pequenas da parte superior do Pad de maneira sobreposta.



Figura 17 – Detalhe interno dos Pads. Placas de alumínio presas à madeira e fios convergindo para furação passante.

Para que o contato atuasse com a pressão da pisada do paciente, a placa de contato precisaria ficar suspensa, sem que as pequenas placas de alumínio de cima encostassem na grande de baixo. Para este fim, um tubo de borracha com diâmetro externo de 6 mm foi afixado com pregos às laterais dos espaços internos dos Pads, conforme a figura 17. O tubo tem resistência suficiente para que sem carga, o Pad não feche contato, e tem elasticidade suficiente para que sob pressão as placas de alumínio da plataforma encostem no terra comum, fechando os contatos. É válido lembrar que não foi testada a resiliência da borracha, então não é possível determinar com este protótipo a vida útil do equipamento bem como o decaimento da sensibilidade dos contatos. Mas o projeto permanece válido para a causa de prova de conceito. Nos testes de sensibilidade foi constatado que parte da borracha precisaria ser removida, pois cobrindo toda a lateral do Pad, era preciso muito peso para que os houvesse fechamento dos contatos.



Figura 18 – Detalhe das borrachas laterais. Foram utilizadas apenas nas laterais mais compridas do Pad de modo que o contato não ficasse tão resistente à pressão.

As placas de alumínio foram pregadas à madeira e em cada placa de alumínio, inclusive as de GND, foram afixados fios de cobre emborrachados, para que o circuito do sensor se completasse. Estes fios saem das placas, se agrupam e seguem através de furos no degrau até as borneiras do primeiro o circuito integrado, que por sua vez se conecta ao Arduino.



Figura 19 – Detalhe da furação feita na estrutura para proteger os fios que saem de dentro dos Pads.

A fim de que os fios que ligam as placas de alumínio ao circuito não interfiram no contato entre as partes, sulcos foram feitos na madeira, a partir da posição de cada placa, até o ponto de agrupamento do conjunto de fios. Isso permitiu a inserção dos fios nos sulcos, e dessa forma os fios não ficam soltos dentro do Pad.

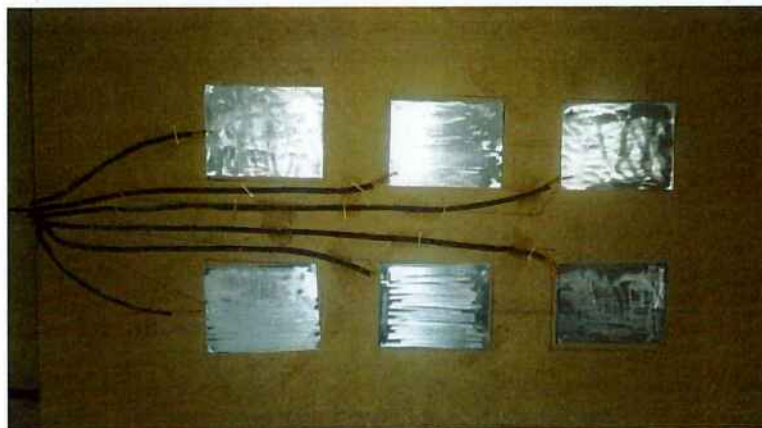


Figura 20 – Detalhe das placas de alumínio dos contatos pregados e os fios encaixados nos sulcos que levam ao ponto da furação passante presos por grampos metálicos.

6.3 Tratamento do sinal de áudio

O sinal de áudio pode ser lido pelo Matlab através de uma função que, ao ler o sinal, o salva em um array, de modo que torna-se possível tratar o sinal. A frequência de amostragem do sinal foi constatada em 44,1kHz, que é uma frequência muito elevada para um sinal simples e por conta disso, o array ficou com aproximadamente 79,5 milhões de pontos, o que é impraticável até mesmo para um computador. Na figura 21 mostra-se um plot do Matlab de parte do sinal original.

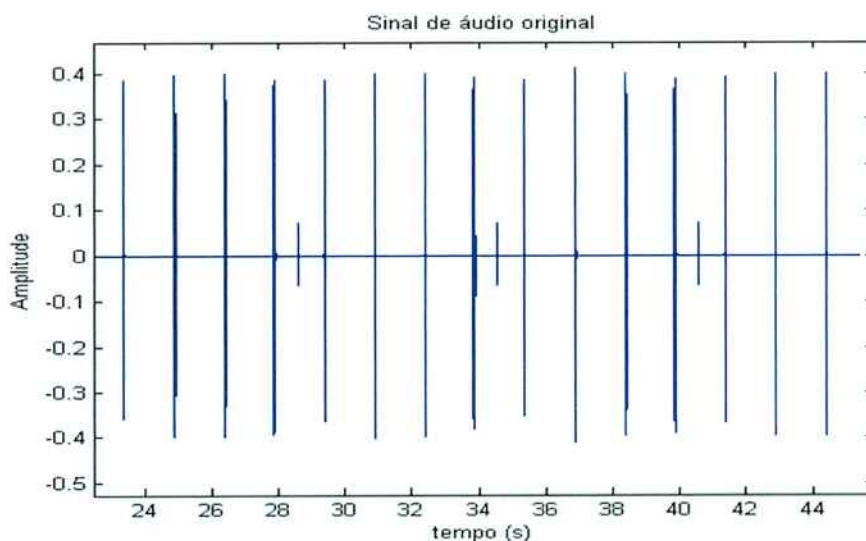


Figure 21 – Sinal de áudio desenhado a partir da leitura feita com função do Matlab.

Um programa em matlab foi escrito para fazer a leitura e tratar o sinal da seguinte forma. Primeiro, era preciso que o sinal lido, fosse modulado, então se aplicou uma rotina de modulação substituindo o valor no array, pelo valor em módulo dividido pelo maior valor em módulo do sinal inteiro. Assim os dados passaram a ser limitados entre zero e um. Mas

somente isso não basta, pois o sinal continua com valores diferentes de zero e um. Logo, fez-se uma filtragem para que valores lidos que estivessem acima de um threshold se tornassem 1 e todos os demais valores abaixo desta cota se tornassem 0. Isso permitiu que o sinal ficasse só com dois valores, porém apesar de parecer ideal, eram muitos pontos e haviam muitos zeros em meio aos uns. Na figura 22 vê-se como ficou o arquivo de áudio após tanto tratamento. Na imagem fica clara a oscilação entre zeros e uns quando há um sinal de estímulo sonoro.

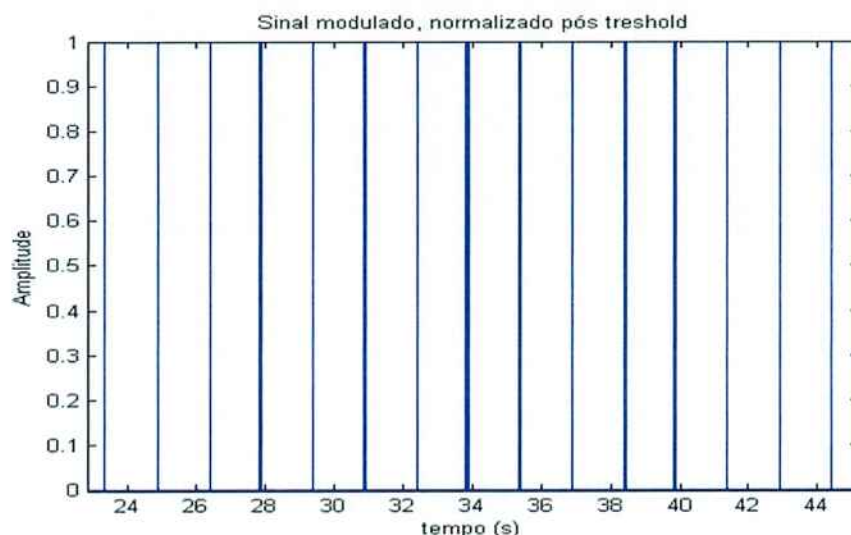


Figure 22 – Sinal pós tratamento de modulação, normalização e lógica com threshold.

A solução desse problema foi criar uma rotina no Matlab que pudesse ler um intervalo do arquivo e decidir se o valor deveria ser zero ou um. Na rotina escrita, o arquivo é lido desde o início e quando uma transição de zero para um é encontrada, um loop condicional inicia um processo de leitura dos 400 pontos à frente e caso a soma dos valores dos pontos for maior que zero, significa que o ponto atual deve ser um. Esta rotina permitiu que o debouncing fosse mitigado, e assim o sinal passou a ser um trem de pulsos com largura variável conforme desejado.

Mas ainda restava o problema do tamanho do arquivo, então foi utilizada uma função de downsampling para reduzir o tamanho do arquivo. A frequência de amostragem foi reduzida de 44,1kHz para apenas 100Hz, reduzindo o arquivo para menos de 100 mil pontos. Na figura 23 mostra-se o sinal após a rotina de debouncing, mas antes do downsampling.

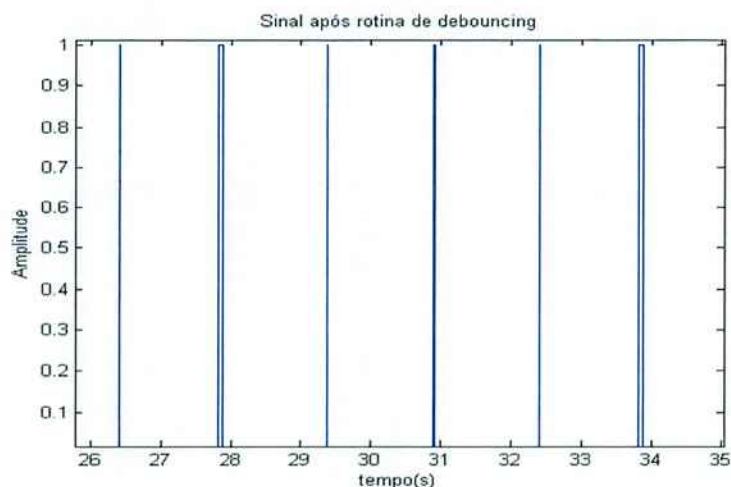


Figure 23 – Trecho em detalhe do sinal pronto antes de passar pela função de downsampling. Neste sinal a sequência de "uns" não é mais permeada de "zeros".

Com o sinal tratado totalmente pela rotina do Apendice A, foi possível reduzir seu tamanho de 1,3Gb para aproximadamente 600kb em formato .txt. Esse arquivo foi então carregado no cartão de memória SD para servir de base para gerar o som do teste através do buzzer.

6.4 Programação do Arduino

Para se reproduzir o sinal de áudio tratado pelo arduino foi utilizado, na parte de circuito, um circuito em protoboard dotado de um Buzzer e, na parte de programação, a biblioteca timer.h para lidar com interrupções internas do microcontrolador. Com essa biblioteca foi possível criar uma sub-rotina que atualiza o estado de uma variável a incrementos constantes de tempo, permitindo que o sinal de áudio seja reproduzido na frequência correta.

Foi utilizado também um shield de leitor de cartão SD, para que fosse possível usar o sinal de áudio completo, com duração de 30 minutos, além de gravar um relatório do teste realizado num documento de texto.

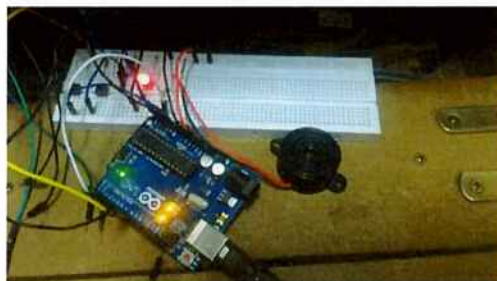


Figure 24 – Detalhe do Arduino conectado ao circuito da protoboard e ao buzzer. Fase de testes em que ainda não havia sido incorporado o Shield de SD.

A programação do Arduino é separada em algumas áreas diferentes. O setup inicia o cartão SD e as funções do microcontrolador, como timers, pinos e a porta serial (usada para debugging).

Durante o setup, é inicializada uma interrupção no timer1 que é associada à função buzz. A função buzz lê de um vetor que foi carregado previamente com a leitura do arquivo de sinal, atualizando o estado do buzzer. Como o interrupt executa a intervalos constantes de tempo, isso garante que o sinal seja executado na frequência correta de amostragem. Na versão final do código, o sinal foi amostrado a 100 Hz e o interrupt executa a cada 100 milissegundos. Paralelamente, essa função conta quantos sinais sonoros foram tocados, de 0 a 3, criando assim 4 estados para o degrau.

Para carregar o vetor que é lido pelo interrupt, foi implementado uma função atualiza_vetor que executa no loop principal da programação. No loop principal, é verificado que posição do vetor de playback está sendo lida. Quando o final do vetor é alcançado, o vetor de playback aponta para o vetor de buffer que estava previamente carregado e o próximo vetor de buffer é carregado. Isso garante que o interrupt sempre vai ler corretamente o vetor carregado com os dados do sinal de áudio. Imediatamente após se carregar o vetor de buffer, é atualizado o arquivo de report com o estado dos pads, do teste e se foi detectado erro ou não, além de se registrar o tempo da gravação.

A detecção de erros foi implementada com rotina condicional que verifica o sinal dos contatos e quando uma configuração errada é identificada, uma variável de erro sinaliza para uma saída analógica do microcontrolador acionar o LED de interface. Na tabela 2 podemos ver a lógica utilizada para determinar quais são as configurações válidas para todos os estados, começando o teste tanto pelo lado direito quanto pelo lado esquerdo.

Tabela 2 – Tabela de configurações válidas para cada estado da programação. Nas colunas "contatos" temos os sinais digitais esperados para cada um dos Pads: 1, 2, 3 e 4 (pela ordem, pads superiores esquerdo e direito e inferiores esquerdo e direito).

Lado Direito							
Estado 0		Estado 1		Estado 2		Estado 3	
Posições válidas	Contatos	Posições válidas	Contatos	Posições válidas	Contatos	Posições válidas	Contatos
3 e 4	1 1 0 0	3 e 4	1 1 0 0	2 e 3	1 0 0 1	1 e 2	0 0 1 1
1 e 4	0 1 1 0	3	1 1 0 1	3	1 1 0 1	1	0 1 1 1
4	1 1 1 0	4	1 1 1 0	2	1 0 1 1	2	1 0 1 1
3	1 1 0 1	2 e 3	1 0 0 1	1 e 2	0 0 1 1	4	1 1 1 0
				1	0 1 1 1	1 e 4	0 1 1 0

Lado Esquerdo							
Estado 0		Estado 1		Estado 2		Estado 3	
Posições válidas	Contatos	Posições válidas	Contatos	Posições válidas	Contatos	Posições válidas	Contatos
3 e 4	1 1 0 0	3 e 4	1 1 0 0	1 e 4	0 1 1 0	1 e 2	0 0 1 1
3	1 1 0 1	3	1 1 0 1	4	1 1 1 0	2	1 0 1 1
4	1 1 1 0	4	1 1 1 0	1	0 1 1 1	1	0 1 1 1
2 e 3	1 0 0 1	1 e 4	0 1 1 0	1 e 2	0 0 1 1	2 e 3	1 0 0 1
				2	1 0 1 1	3	1 1 0 1

6.5 Montagem do circuito em Protoboard

Foram montadas duas protoboard com circuitos analógicos para receber o sinal dos Pads. A primeira foi afixada na parte superior dentro do degrau, longe de qualquer contato com o usuário, e permitindo a concentração de todos os fios provenientes dos quatro Pads. Já a segunda foi instalada junto ao Arduino do lado de fora, para ser acessada e configurar o teste.

6.5.1 Protoboard interna

O propósito da protoboard afixada internamente no degrau é concentrar todos os fios advindos dos Pads, e fazer a parte do circuito de pull-up que consiste o contato elétrico do Pad, conforme a imagem da figura 24:



Figura 25 – Esquema elétrico Pull-up.

Foram utilizadas borneiras próprias para circuitos em protoboard para que os fios dos contatos pudessem ser conectados ao sistema, e o valor da resistência utilizada para cada Pad foi de 5k Ω . Na figura 25 é possível ver o circuito montado, esperando pela conexão dos fios.

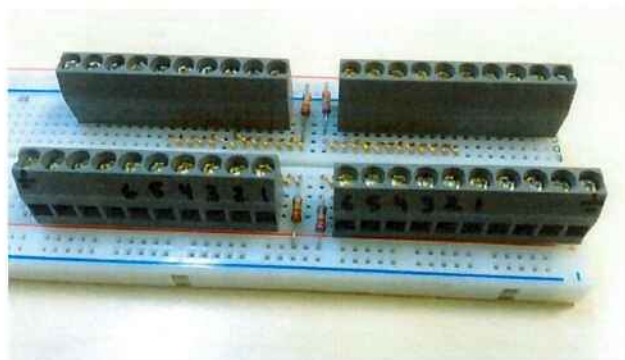


Figura 26 – Circuito de interface entre fios dos Pads e sistema com Arduino. Detalhe para a numeração dos contatos de 1 a 6 e para o sinal de terra comum.

A protoboard foi instalada na margem dianteira do degrau, de forma que as borneiras dos Pads 3 e 4 pudessem ser acessadas pela frente. Isto é importante para o uso do degrau, já que é preciso desconectar essas borneiras para que o degrau possa ser desmontado e guardado retraído.

Abaixo na figura 26 pode-se ver detalhe da protoboard interna, ainda não afixada à madeira, mas já conectada aos fios dos Pads.

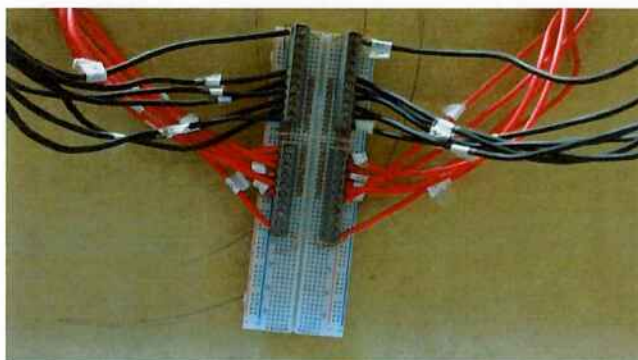


Figure 27 – Protoboard interna conectada aos fios dos Pads. Convencionou-se que os fios vermelhos são advindos dos Pads superiores (1 e 2) e os fios pretos são advindos dos Pads inferiores (3 e 4).

6.5.2 Protoboard externa

O circuito fixado na parte externa, montado num protoboard, consiste de o Arduino, o shield de cartão SD, o buzzer e alguns botões e LEDs de interface.

Tanto o arduino quanto o shield precisam estar facilmente acessíveis para trocar o cartão SD, se necessário e conectar o Arduino a um laptop. Os botões e LEDs também precisavam estar acessíveis, já que funcionam como interface. Os botões funcionam para iniciar e parar o teste, além de escolher se o paciente irá começar com a perna direita ou esquerda. O LED acusa se foi detectado um erro no teste atual.

7 VERIFICAÇÃO DO PROJETO

A fase de testes se dividiu em três etapas de teste: teste de contato, teste de sensibilidade, e teste de funcionamento. A primeira e a segunda começaram ainda na montagem dos Pads, com verificação constante a cada etapa da montagem para garantir que os contatos permanecessem funcionais. Já a segunda foi executada com o degrau em sua fase final de montagem.

7.1 Teste de contato

Após a montagem dos Pads, e passagem dos fios através dos furos passantes, foi executada uma bateria de testes para verificar se havia contato no Pad quando este era pisado. Esse teste foi feito em duas etapas utilizando um multímetro de mão. Primeiro, o multímetro foi utilizado para detectar curto circuito, isto é contato, quando houvesse pressão sobre o Pad. O aparelho é capaz de medir a resistência e emitir um sinal sonoro agudo quando um curto é detectado.

Segundo, foi feito um teste aplicando-se tensão de 5 V nas placas superiores de cada Pad e aterrando as placas inferiores. Para isso foi utilizado o circuito Pull-up já montado na protoboard e a alimentação do próprio Arduino. O teste consistia em verificar se a tensão de 5 V presente com o Pad estava em repouso caía para zero quando o Pad era pressionado. Isso implicaria que o contato havia sido fechado como um curto e toda a tensão estaria se aplicando sobre o resistor.

7.2 Testes de Sensibilidade

Para testar a sensibilidade dos Pads, foi necessário a ajuda de uma pessoa mais magra, para que se garantisse que qualquer paciente pudesse executar o teste sem o perigo de resultados falsos. Assim, a pessoa convidada auxiliou em uma série de testes em todos os contatos, a saber:

- manter todo o peso em apenas um Pad;
- distribuir todo o peso apenas nos Pads inferiores; repetiu-se para os Pads superiores;
- Posicionar os pés sobre os três pares de placas de alumínio dos contatos (1 e 2, 3 e 4, 5 e 6) para que pudesse ser verificada a influencia da posição do pé sobre o Pad.

7.3 Testes Gerais

Antes de o degrau ficar pronto, testes ao longo de sua montagem foram feitos com a programação em Arduino, para que o processo de implementação pudesse ser executado. Na figura 27 mostra-se um circuito primordial usado para testar a programação, no qual os botões simulavam os contatos e os LEDs sinalizavam o estado.

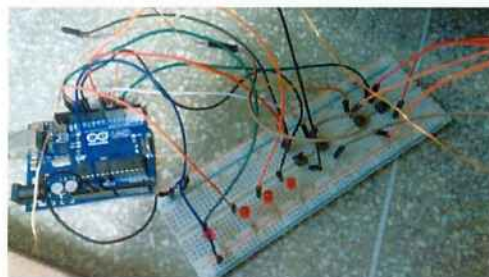


Figura 28 – Circuito de teste para debugging da programação em Arduino.

Com todo o sistema montado e operante foi possível começar os testes com os integrantes do grupo e pessoas convidadas a ajudar. Estes testes internos foram feitos seguindo o protocolo de aplicação do exame, mas nenhum teste do degrau completo foi aplicado em razão de se economizar tempo de execução. Nas imagens a seguir, figura 28, mostra-se as posições dos quatro estados que o paciente passa para completar um ciclo do teste.



Figure 29 – Posições do paciente ao executar o Teste do Degrau. Da esquerda para a direita vê-se a progressão do teste iniciado com o pé direito.

O teste final serviria para apontar os erros na sensibilidade dos contatos, na programação do Arduino, e para verificar se os sinais gerados pelo Arduino eram compatíveis com o arquivo de áudio original.

7.4 Resultados dos testes

Em linhas gerais, os testes de contato e de sensibilidade foram importantes para a revisão do número de placas de alumínio instaladas nos Pads, pois a princípio verificou-se que os Pads estavam em quase todos os contatos com uma sensibilidade muito baixa. Isso resultou na redução da quantidade de tubos de borracha e em seguida uma nova bateria de testes.

A sensibilidade baixa persistiu, então tomou-se a decisão de acrescentar placas de alumínio extras nos contatos de modo a aproximá-los. Primeiro na parte superior, e

após mais testes, aplicou-se também na parte inferior. Por fim, a sensibilidade dos Pads ficou na medida exata para a aplicação do teste geral.

Os testes finais foram essenciais para que erros fossem encontrados, tanto na montagem do degrau, quanto na programação do Arduino. Foi constatado mau contato em alguns jumpers, que foram substituídos resolvendo o problema. Além disso, ao testar o degrau começando com o pé direito e esquerdo, pode-se verificar e corrigir erros na programação de detecção de erros.

O degrau, no final dos testes, desempenhou conforme desejado, e assim fica pronto para receber testes reais com pacientes das colegas fisioterapeutas da São Camilo.

8 DISCUSSÕES E CONCLUSÃO

O projeto foi um sucesso, sendo possível atender as especificações da equipe de fisioterapeutas e tendo sido a prova de conceito e os testes preliminares bem sucedidos. Como esse projeto vai se estender além do TCC da turma de 2016, será necessário dar um acabamento final no circuito do degrau, imprimindo a versão final, além de adicionar uma fonte para alimentação do microcontrolador.

REFERÊNCIAS

ANDRADE, Carlos H. S.; CIANCI, Reinaldo G.; MALAGUTI, Carla; CORSO, Simone D.. O uso de testes do degrau para a avaliação da capacidade de exercício em pacientes com doenças pulmonares crônicas. J Bras Pneumol. v. 38, n. 1, p. 116-124, 2011.

MARRARA, Kamilla T.; MARINO, Diego M.; JAMAMI, Maurício; JUNIOR, Antônio D. O.; DI LORENZO, Valéria A. P.. Responsividade do teste do degrau de seis minutos a um programa de treinamento físico em pacientes com DPOC. J Bras Pneumol. v. 38, n. 5, p. 579-587, 2012.

ANDRADE, Gabriela S.; FIGUEIREDO, Thamyres. Avaliação da capacidade física de indivíduos saudáveis e portadores de DPOC por meio de novo protótipo para o teste do degrau. 2016. 33f. Projeto de Pesquisa (TCC) – Centro Universitário São Camilo, São Paulo, 2016

APÊNDICE A

Abaixo mostra-se a rotina em matlab que foi escrita para o tratamento do sinal de áudio.

```
%% script para converter .wav em .txt%%
```

```
% Marcelo Perillo
```

```
%% arquivo de audio deve estar no diretório dos arquivos do Matlab %%
```

```
clear all
```

```
close all
```

```
[A, fs] = audioread('degrau.wav');
```

```
T = 1/fs;
```

```
ts = 0:T:1804;
```

```
n = length(A);          % normalização
```

```
m = abs(max(A));
```

```
B = A/m;
```

```
for r = 1:n
```

```
    if B(r) < 0.5          % gera uns e zeros
```

```
        B(r) = 0;
```

```
    else B(r) = 1;
```

```
    end
```

```
end
```

```
for r = 2:n-400          % corrige debouncing
```

```
    if B(r) == 1 || B(r-1) == 1
```

```
        if sum(B(r+1:r+400)) ~= 0
```

```
            B(r+1) = 1;
```

```
        end
```

```
    end
```

```

end
%% salva em txt
% fileID = fopen('sinal.txt','w');
% string = sprintf('%d\r\n',Z);
% fprintf(fileID,string);
% fclose(fileID);
%% aquisição de período da janela do sinal
t = 1;
J = zeros(1,4681);
for r = 2:n
    if B(r-1) == 0 && B(r) == 1
        J(t) = r/44100;
        t = t+1;
    end
end
K = zeros(1,1170);
for r = 0:1170
    K(r+1)=J(1+4*r);
end
delta = zeros(1,1170);
for r = 1:1170
    delta(r) = K(r+1)-K(r);
end
%vetores provenientes do Excel

```



```
a=[1 11 23 37 53 71 91 113 137 163 191 221 253 287 323 361 401 443 487 533 581  
631 683 737 793 851 911 973 1037 1103];  
  
b=[10 22 36 52 70 90 112 136 162 190 220 252 286 322 360 400 442 486 532 580 630  
682 736 792 850 910 972 1036 1102 1170];  
  
for r = 1:30  
  
    M(r) = mean(delta(a(r):b(r)));  
  
end
```

APÊNDICE B

Abaixo mostra-se a programação feita no Arduino que foi escrita para tocar o sinal de áudio tratado pelo Matlab através do buzzer.

```
void buzz()

{

int saida;

if (flag_go == 0){                                //flag que sinaliza se esta ocorrendo playback

    blinkCount = 0;                                //contador de quantas casas do vetor já foram lidas

    sinal = 0;                                    // contador de quantas vezes foi tocado o sinal sonoro

}

blinkCount = blinkCount +1; //incrementa contador da posição do vetor

if (blinkCount > 29) blinkCount = 0;                //verifica se o vetor atual já foi lido

oldLed = ledState; // armazena estado anterior da saida

if(flag_vetor == 1) saida = vetor1[blinkCount]; //verifica qual vetor deve ser

if(flag_vetor == -1) saida = vetor2[blinkCount]; //lido

digitalWrite(buzz, saida); //atualiza estado do buzzer

if (saida == 1) ledState = ON;

if (saida == 0) ledState = OFF;

int fvetor_old = flag_vetor;

if (blinkCount == 29) flag_vetor = flag_vetor*-1; //troca vetor caso seja ultima leitura

if(fvetor_old != flag_vetor) flag_atualiza = flag_atualiza*-1; //aciona flag sinalizando

//que proximo vetor                //deve ser carregado

if (ledState == ON){ //atualiza quantas vezes o sinal sonoro foi tocado

    if(oldLed == OFF){

        sinal = sinal+1;

    }

}

if (sinal>3) sinal =0; //garante que so existam 4 incrementos estados do sinal sonoro

}
```

APÊNDICE C

Abaixo mostra-se a programação feita no Arduino que foi escrita para o processamento dos dados advindos dos Pads e monitoração do teste.

```
void setup()
{
    Timer1.initialize();

    Timer1.initialize(10000); //periodo em micro s //

    Timer1.attachInterrupt(buzz);

    Serial.begin(9600);

    Serial.print("Initializing SD card..."); //inicializa SD

    pinMode(10, OUTPUT);

    if (!SD.begin(4)) {
        Serial.println("initialization failed!");
        return;
    }
    Serial.println("initialization done.");

    pos = atualiza_vetor(0, vetor1); //carrega primeiro vetor saida
}

void loop()
{
    int pad1 = digitalRead(bot1); // le pads
    int pad2 = digitalRead(bot2); // le pads
    int pad3 = digitalRead(bot3); // le pads
    int pad4 = digitalRead(bot4); // le pads

    if (!digitalRead(BUTTON3)){ //le qual lado o paciente esta usando
        lado = 3; //direito
    }

    if (!digitalRead(BUTTON2)){
```

```

lado = 2; //esquerdo
}

//checa se o estado dos pads atual eh estado de erro. Maquina de estados depende de quantos
//sinais sonoros foram tocados (clock da maquina de estados) e de qual lado o paciente iniciou o
teste

if (lado == 2){//esquerdo

    if (sinal==0){

        if (pad1==HIGH && pad2==HIGH && pad3==HIGH && pad4==HIGH) erro =1;

        //if (pad1==HIGH && pad2==HIGH && pad3==HIGH && !(pad4==HIGH)) erro =1;

        if (pad1==HIGH && pad2==HIGH && !(pad3==HIGH) && pad4==HIGH) erro =1;

        //if (pad1==HIGH && pad2==HIGH && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

        if (pad1==HIGH && !(pad2==HIGH) && pad3==HIGH && pad4==HIGH) erro =1;

        //if (pad1==HIGH && !(pad2==HIGH) && pad3==HIGH && !(pad4==HIGH)) erro =1;

        if (pad1==HIGH && !(pad2==HIGH) && !(pad3==HIGH) && pad4==HIGH) erro =1;

        if (pad1==HIGH && !(pad2==HIGH) && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

        if (!(pad1==HIGH) && pad2==HIGH && pad3==HIGH && pad4==HIGH) erro =1;

        if (!(pad1==HIGH) && pad2==HIGH && pad3==HIGH && !(pad4==HIGH)) erro =1;

        if (!(pad1==HIGH) && pad2==HIGH && !(pad3==HIGH) && pad4==HIGH) erro =1;

        if (!(pad1==HIGH) && pad2==HIGH && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

        if (!(pad1==HIGH) && !(pad2==HIGH) && pad3==HIGH && pad4==HIGH) erro =1;

        if (!(pad1==HIGH) && !(pad2==HIGH) && pad3==HIGH && !(pad4==HIGH)) erro =1;

        if (!(pad1==HIGH) && !(pad2==HIGH) && !(pad3==HIGH) && pad4==HIGH) erro =1;

        if (!(pad1==HIGH) && !(pad2==HIGH) && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

    }

    if (sinal==1){

        if (pad1==HIGH && pad2==HIGH && pad3==HIGH && pad4==HIGH) erro =1;

        if (pad1==HIGH && pad2==HIGH && pad3==HIGH && !(pad4==HIGH)) erro =1;

        //if (pad1==HIGH && pad2==HIGH && !(pad3==HIGH) && pad4==HIGH) erro =1;

        //if (pad1==HIGH && pad2==HIGH && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

        if (pad1==HIGH && !(pad2==HIGH) && pad3==HIGH && pad4==HIGH) erro =1;

        if (pad1==HIGH && !(pad2==HIGH) && pad3==HIGH && !(pad4==HIGH)) erro =1;

        if (pad1==HIGH && !(pad2==HIGH) && !(pad3==HIGH) && pad4==HIGH) erro =1;

        if (pad1==HIGH && !(pad2==HIGH) && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

    }

```



```

//if (pad1==HIGH && !(pad2==HIGH) && pad3==HIGH && pad4==HIGH) erro =1;
//if (pad1==HIGH && !(pad2==HIGH) && pad3==HIGH && !(pad4==HIGH)) erro =1;
if (pad1==HIGH && !(pad2==HIGH) && !(pad3==HIGH) && pad4==HIGH) erro =1;
if (pad1==HIGH && !(pad2==HIGH) && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;
if (!(pad1==HIGH) && pad2==HIGH && pad3==HIGH && pad4==HIGH) erro =1;
if (!(pad1==HIGH) && pad2==HIGH && pad3==HIGH && !(pad4==HIGH)) erro =1;
if (!(pad1==HIGH) && pad2==HIGH && !(pad3==HIGH) && pad4==HIGH) erro =1;
if (!(pad1==HIGH) && pad2==HIGH && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;
//if (!(pad1==HIGH) && !(pad2==HIGH) && pad3==HIGH && pad4==HIGH) erro =1;
if (!(pad1==HIGH) && !(pad2==HIGH) && pad3==HIGH && !(pad4==HIGH)) erro =1;
if (!(pad1==HIGH) && !(pad2==HIGH) && !(pad3==HIGH) && pad4==HIGH) erro =1;
if (!(pad1==HIGH) && !(pad2==HIGH) && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;
}
}

```

```

if (lado == 3){ //direito
    if (sinal==0){
        if (pad1==HIGH && pad2==HIGH && pad3==HIGH && pad4==HIGH) erro =1;
        if (pad1==HIGH && pad2==HIGH && pad3==HIGH && !(pad4==HIGH)) erro =1;
        //if (pad1==HIGH && pad2==HIGH && !(pad3==HIGH) && pad4==HIGH) erro =1;
        //if (pad1==HIGH && pad2==HIGH && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;
        if (pad1==HIGH && !(pad2==HIGH) && pad3==HIGH && pad4==HIGH) erro =1;
        if (pad1==HIGH && !(pad2==HIGH) && pad3==HIGH && !(pad4==HIGH)) erro =1;
        if (pad1==HIGH && !(pad2==HIGH) && !(pad3==HIGH) && pad4==HIGH) erro =1;
        if (pad1==HIGH && !(pad2==HIGH) && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;
        if (!(pad1==HIGH) && pad2==HIGH && pad3==HIGH && pad4==HIGH) erro =1;
        if (!(pad1==HIGH) && pad2==HIGH && pad3==HIGH && !(pad4==HIGH)) erro =1;
        //if (!(pad1==HIGH) && pad2==HIGH && !(pad3==HIGH) && pad4==HIGH) erro =1;
        if (!(pad1==HIGH) && pad2==HIGH && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;
        if (!(pad1==HIGH) && !(pad2==HIGH) && pad3==HIGH && pad4==HIGH) erro =1;
        if (!(pad1==HIGH) && !(pad2==HIGH) && pad3==HIGH && !(pad4==HIGH)) erro =1;
    }
}

```



```

    if (!(pad1==HIGH) && pad2==HIGH && !(pad3==HIGH) && pad4==HIGH) erro =1;

    if (!(pad1==HIGH) && pad2==HIGH && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

    //if (!(pad1==HIGH) && !(pad2==HIGH) && pad3==HIGH && pad4==HIGH) erro =1;

    if (!(pad1==HIGH) && !(pad2==HIGH) && pad3==HIGH && !(pad4==HIGH)) erro =1;

    if (!(pad1==HIGH) && !(pad2==HIGH) && !(pad3==HIGH) && pad4==HIGH) erro =1;

    if (!(pad1==HIGH) && !(pad2==HIGH) && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

}

if (sinal==3) {

    if (pad1==HIGH && pad2==HIGH && pad3==HIGH && pad4==HIGH) erro =1;

    if (pad1==HIGH && pad2==HIGH && pad3==HIGH && !(pad4==HIGH)) erro =1;

    if (pad1==HIGH && pad2==HIGH && !(pad3==HIGH) && pad4==HIGH) erro =1;

    if (pad1==HIGH && pad2==HIGH && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

    if (pad1==HIGH && !(pad2==HIGH) && pad3==HIGH && pad4==HIGH) erro =1;

    if (pad1==HIGH && !(pad2==HIGH) && pad3==HIGH && !(pad4==HIGH)) erro =1;

    if (pad1==HIGH && !(pad2==HIGH) && !(pad3==HIGH) && pad4==HIGH) erro =1;

    if (pad1==HIGH && !(pad2==HIGH) && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

    //if (!(pad1==HIGH) && pad2==HIGH && pad3==HIGH && pad4==HIGH) erro =1;

    if (!(pad1==HIGH) && pad2==HIGH && pad3==HIGH && !(pad4==HIGH)) erro =1;

    //if (!(pad1==HIGH) && pad2==HIGH && !(pad3==HIGH) && pad4==HIGH) erro =1;

    if (!(pad1==HIGH) && pad2==HIGH && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

    //if (!(pad1==HIGH) && !(pad2==HIGH) && pad3==HIGH && pad4==HIGH) erro =1;

    if (!(pad1==HIGH) && !(pad2==HIGH) && pad3==HIGH && !(pad4==HIGH)) erro =1;

    if (!(pad1==HIGH) && !(pad2==HIGH) && !(pad3==HIGH) && pad4==HIGH) erro =1;

    if (!(pad1==HIGH) && !(pad2==HIGH) && !(pad3==HIGH) && !(pad4==HIGH)) erro =1;

}

}

```

```
digitalWrite(led1, erro); //acende o led se ocorreu erro
```

```
fgo_old = flag_go; // armazena estado da flag anterior
```

```
if (!digitalRead(BUTTON1)) flag_go = 1; //verifica se foi iniciado um teste
```

```
if (!digitalRead(BUTTON4)) flag_go = 0;
```

```

if (fgo_old != flag_go) reinicia = 1; //armazena se teste foi reiniciado

//atualiza vetores
if (flag_atualiza == -1 && flag_go == 1) { //verifica se esta ocorrendo teste e precisa atualizar
    flag_atualiza = 1; //vetor buffer
    if (flag_vetor == 1) pos = atualiza_vetor(pos, vetor2); //atualiza vetor buffer
    if (flag_vetor == -1) pos = atualiza_vetor(pos, vetor1); //atualiza vetor buffer
}

if(flag_go == 0 && reinicia == 1) { //verifica se o teste foi reiniciado
    pos = atualiza_vetor(0, vetor1); //reinicia vetor buffer
    flag_atualiza = -1; //marca que não eh mais necessário reiniciar teste
}
}

void buzz()
{
    int saida;

    if (flag_go == 0){ //flag que sinaliza se esta ocorrendo playback
        blinkCount = 0; //contador de quantas casas do vetor já foram lidas
        sinal = 0; // contador de quantas vezes foi tocado o sinal sonoro
    }

    blinkCount = blinkCount +1; //incrementa contador da posição do vetor
    if (blinkCount > 29) blinkCount = 0; //verifica se o vetor atual já foi lido
    oldLed = ledState; // armazena estado anterior da saida

    if(flag_vetor == 1) saida = vetor1[blinkCount]; //verifica qual vetor deve ser
    if(flag_vetor == -1) saida = vetor2[blinkCount]; //lido
    digitalWrite(buzz, saida); //atualiza estado do buzzer

    if (saida == 1) ledState = ON;

```

```

if (saida == 0) ledState = OFF;

int fvetor_old = flag_vetor;

if (blinkCount == 29) flag_vetor = flag_vetor*-1; //troca vetor caso seja ultima leitura
if(fvetor_old != flag_vetor) flag_atualiza = flag_atualiza*-1; //aciona flag sinalizando
//que proximo vetor          //deve ser carregado

if (ledState == ON){ //atualiza quantas vezes o sinal sonoro foi tocado
    if(oldLed == OFF){
        sinal = sinal+1;
    }
}

if (sinal>3) sinal =0; //garante que so existam 4 incrementos estados do sinal sonoro
}

long atualiza_vetor(long posicao, int vetor[30]){
    char chara;
    String str = "";
    myFile = SD.open("sinal.txt"); //abre o arquivo de sinal para leitura
    pos = 0;
    line = 0;
    if (myFile){
        //se o arquivo esta disponivel
        myFile.seek(3*posicao);
        //avanca para a proxima posicao a ser lida
        while (line <= 29){
            //enquanto nao foi lido a extensao total de um vetor
            pos = myFile.position();
            //atualiza a posicao atual
            if(3*posicao+3*line<= myFile.size()){
                //verifica se está dentro do arquivo
                chara = myFile.read();
                //le a proxima posicao do arquivo
                str = str + chara;
                //armazena o caractere lido num string
                val = str.toInt();
                //converte o string para int
                myFile.read();
                //pula dois caracteres especiais
                myFile.read();
                //

```

```

    }

    else val = 0;

    str = ""; //reinicia a string para a proxima leitura
    vetor[line]=val; //atualiza a posicao atual do vetor buffer
    line += 1; //atualiza a linha sendo lida
  }
}

else Serial.println("Arquivo indisponivel."); //caso arquivo nao esteja disponivel

myFile.close(); //fecha arquivo
return line + posicao; //retorna a posicao em que o arquivo esta sendo lido
}

```