

**INSTITUTO ENSINAR BRASIL
FACULDADES DOCTUM DE CARATINGA**

RAFAEL FELIPE REIS ALTINO

**O USO DE ARDUINO NA ELABORAÇÃO DE UMA
FERRAMENTA INCLUSIVA DIGITAL PARA PESSOAS COM
DEFICIÊNCIA MOTORA**

CARATINGA





2018

RAFAEL FELIPE REIS ALTINO
FACULDADES DOCTUM DE CARATINGA

**O USO DE ARDUINO NA ELABORAÇÃO DE UMA
FERRAMENTA INCLUSIVA DIGITAL PARA PESSOAS COM
DEFICIÊNCIA MOTORA**

Monografia apresentada à banca examinadora da Faculdade de Ciência da Computação das Faculdades Doctum de Caratinga como exigência parcial para obtenção do grau de bacharel em Ciência da Computação, sob orientação do professor Msc. Ricardo Botelho Campos.

DOCTUM/CARATINGA
18/12/2018

	FACULDADES DOCTUM DE CARATINGA	FORMULÁRIO 9
	TRABALHO DE CONCLUSÃO DE CURSO	
TERMO DE APROVAÇÃO		
TERMO DE APROVAÇÃO		
<p>O Trabalho de Conclusão de Curso intitulado: O USO DE ARDUINO NA ELABORAÇÃO DE UMA FERRAMENTA INCLUSIVA DIGITAL PARA PESSOAS COM DEFICIÊNCIA MOTORA, elaborado pelo(s) aluno(s) RAFAEL FELIPE REIS ALTINO foi aprovado por todos os membros da Banca Examinadora e aceito pelo curso de CIÊNCIA DA COMPUTAÇÃO das FACULDADES DOCTUM DE CARATINGA, como requisito parcial da obtenção do título de</p>		
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.		
Caratinga 18/12/2018		
 _____ RICARDO BOTELHO CAMPOS Prof. Orientador		
 _____ ELIAS DE SOUZA GONÇALVES Prof. Avaliador 1		
 _____ MAICON RIBEIRO Prof. Examinador 2		

AGRADECIMENTOS

Primeiramente agradeço a Deus, por até aqui ter me concedido o prazer da vida e por me possibilitar estar concluindo esta etapa de minha vida, que acredito eu, será apenas o início de várias outras.

Agradeço a todos aos meus familiares e amigos, que me ajudaram a concluir este propósito, através de orações e apoio, foram de grande suporte. Gostaria de agradecer em especial aos meus pais Rafael Altino e Andréa Christina Reis Altino que desempenharam um importante papel na trajetória deste trabalho, me apoiaram e incentivaram, tudo que conquistei hoje devo a vocês. Gostaria de agradecer também a minha tia Alessandra Reis que se prontificou a dedicar parte de sua apertada rotina para me ajudar na conclusão do meu projeto. Aos meus amigos Cydweinner Gonçalves, Lívia Maia e Thiago Neves, desejo todo o sucesso possível e agradeço por me proporcionarem inesquecíveis lembranças nestes anos de curso.

Gostaria de agradecer aos meus professores, que sempre se dedicaram para nos garantir um futuro melhor através da educação. Cada um que, diferente do outro, me ajudou a enxergar o mundo das mais variadas formas. E me fizeram perceber que isto não é um adeus, mas apenas um Hello World!

RESUMO

Este trabalho consiste na elaboração de uma ferramenta denominada Blackbox com a função de captar movimentos através de contato no queixo do usuário e a partir destes estímulos, movimentar o cursor do mouse, possibilitando assim, de forma parcial, o uso do computador sem a necessidade do uso das mãos. Foram realizados estudos em livros que tratam o processo de programação utilizada na plataforma Arduino, recursos inclusivos, além de buscas por monografias e trabalhos voltados para a área de conhecimento dos diversos tipos de paralisias e limitações motoras de membros superiores. Este projeto é direcionado para pessoas que possuem tetraplegia ou alguma outra espécie de deficiência, que limite ou impeça os movimentos dos braços do usuário. O projeto tem como microcontrolador um ATmega 328, sendo este, um produto de baixo custo em comparação a outros microcontroladores.

Palavras chaves: Deficiência motora. Arduino. Inclusão.

ABSTRACT

This work consists in the elaboration of a tool called Blackbox with the function of capturing movements through contact in the user's chin and from these stimuli, to move the mouse cursor, thus allowing, in a partial way, the use of the computer without the necessity of the use of hands. Studies were carried out in books that deal with the programming process used in the Arduino platform, inclusive resources, as well as searches for monographs and papers focused on the area of knowledge of the various types of paralyzes and motor limitations of upper limbs. This project is aimed at people who have tetraplegia or some other kind of disability that limits or impedes the movements of the user's arms. The project has as microcontroller an ATmega 328, this being a product of low cost compared to other microcontrollers.

Keywords: Motor Disability. Arduino. Inclusion.

LISTA DE ILUSTRAÇÕES

Figura 1. Arduino Uno.....	16
Figura 2. Medula Espinhal.....	19
Figura 3. Principais causas para traumas medulares.....	21
Figura 4. Arduino IDE.....	26
Figura 5. Netbeans IDE.....	27
Figura 6. A caixa.....	28
Figura 7. Apoio e extensor.....	29
Figura 8. Suporte.....	29
Figura 9: Teste da Calculadora.....	30
Figura 10: Teste do texto.....	31

LISTA DE TABELAS

Tabela 1. Componentes físicos usados.....	24
Tabela 2. Programas usados.....	24
Tabela 3. Tempo de duração por teste.....	32

LISTA DE SIGLAS

API – Application Programming Interface
FTDI – Future Technology Devices International
IDE – Integrated Development Environment
JNA – Java Native Access
LDR – Light Dependent Resistor
MSD – Merck Sharp and Dohme
TRM – Traumatismo Raquimedular
USB – Universal Serial Bus

SUMÁRIO

1. INTRODUÇÃO	12
2. REFERENCIAL TEÓRICO	14
2.1 Microcontroladores	14
2.2 Arduino	15
2.2.1 Arduino Uno	16
2.3 Sensores	17
2.3.1 Sensores Digitais	17
2.3.2 Sensores Analógicos.....	18
2.4 Módulos para Arduino	18
2.4.1 Joystick Analógico.....	18
2.5 Deficiência motora.....	19
2.6 Lesão medular.....	19
2.6.1 Etiologia traumática.....	20
2.6.2 Etiologia não traumática.....	21
2.7 Tetraplegia	21
3. METODOLOGIA.....	23
3.1 Objeto de estudo	23
3.2 Componentes	23
3.2.1 Montagem	23
3.3 Sketch	25
3.4 Desenvolvimento do suporte	23
3.5 Realização dos testes	23
3.6 Análise dos resultados	31
3.7 Desafios encontrados.....	32
3.8 Possíveis aprimoramentos	33

3.9 Trabalhos futuros	33
4. CONCLUSÃO.....	34
5. REFERÊNCIAS.....	36
6. ANEXO A – CÓDIGOS FONTE UTILIZADOS	36

1. INTRODUÇÃO

A cada dia mais, a sociedade contemporânea possibilita acesso de pessoas aos computadores e a internet, com isto, eles têm se tornado ferramentas comuns do cotidiano, assim proporcionando a realização de ações de forma mais eficiente, como a comunicação, através de mensagens de texto e vídeo-chamadas, formas bem mais rápidas na troca de informações, se em comparação ao telefone e a troca de cartas que eram alguns dos recursos adotados, há alguns anos atrás.

Os novos recursos contribuíram também para a globalização e a ampliação do acesso à informação, possibilitando as pessoas a lerem notícias pela internet na hora em que elas desejassem além de possibilitar o acesso a informações de qualquer lugar do mundo, inclusive em tempo real. Não é apenas na troca de informações que os computadores têm desempenhado um papel importante, pois o avanço tecnológico também tem se mostrado eficiente em várias áreas do conhecimento, pois o computador é uma máquina que veio revolucionar, por meio de sua rapidez, organização e confiabilidade que ele permite. Sua grande importância nos dias de hoje com essa era tecnológica, possibilita uma economia de tempo e ganho na qualidade, o que facilita a vida de qualquer pessoa.

Com o passar dos anos, as novas descobertas científicas com a utilização de recursos tecnológicos, possibilitaram o aprimoramento na área da medicina, na sofisticação quanto a produção de remédios, na automação e agilidade dos atendimentos em geral, entre outros serviços. Mesmo com facilidade para a aquisição de recursos tecnológicos, em especial o computador, algumas pessoas não conseguem utilizá-los por diversos tipos de empecilhos, entre eles a falta de recursos financeiros, ausência de conhecimento, entre outras questões. Existem grandes barreiras, que nem sempre recebem a atenção merecida, destas, algumas estão diretamente ligadas às barreiras que se apresentam nos usuários acometidos por limitações físicas, mentais ou sensoriais. Sabe-se que as limitações por menores que sejam, podem impedir uma pessoa de alcançar com igualdade de oportunidades o acesso pleno ao conhecimento ou a qualquer outro fim, que se deseja alcançar.

Há alguns anos, por meio de políticas públicas específicas, ouve-se falar

discretamente e com pouca notoriedade a respeito da inclusão de pessoas com deficiência nos diferentes espaços sociais. Muitas são as dificuldades, entre elas a falta de acessibilidade. Neste sentido, propõe-se na execução desse projeto, o recurso do uso de uma ferramenta, que possibilitará entre outras incapacidades a utilização do mesmo por pessoa com limitações motoras afins, a utilização do computador de forma funcional e independente.

Algumas ferramentas eficientes foram criadas para minimizar a dificuldade de uso do computador por pessoas com deficiência, dentre eles alguns gratuitos, como por exemplo softwares como o Dosvox, Jaws e Orca que realizam a leitura para deficientes visuais. Entretanto, alguns recursos físicos, como o computador com linha braile (com preço médio entre R\$ 10.000,00 a R\$ 20.000,00) e mouses criados para pessoas com deficiência motora são de alto custo ou são poucos vendidos no mercado, o que dificulta o acesso para pessoas que dependem ou poderiam beneficiar-se deles. Sendo assim, pretende-se com a criação da Blackbox, uma ferramenta capaz de oportunizar o acesso a um maior número de pessoas, por meio de um recurso eficiente, funcional e, sobretudo acessível financeiramente.

Com este projeto ainda de forma experimental, pretende-se apresentar a demonstração com resultados satisfatórios de uma ferramenta que poderá facilitar maior emancipação, autonomia e melhorar a qualidade de vida de deficientes, numa rotina diária e mais acessível, e com a facilidade de ser um recurso de baixo custo.

2. REFERENCIAL TEÓRICO

As seções a seguir contextualizam os principais temas que serão abordados durante o trabalho, apresentando os conceitos mais importantes dentro de cada tópico.

2.1 Microcontroladores

Os microcontroladores reúnem vários recursos em um só circuito interligado de pequenas proporções, sendo estes recursos: um microprocessador, as memórias, pinos de entrada e saída dentre outros. Martins descreve de forma mais detalhada estes componentes:

Tipicamente, um microcontrolador caracteriza-se por incorporar no mesmo encapsulamento um microprocessador (com a finalidade de interpretar as instruções de programa e processar dados), memória de programa (com a finalidade de armazenar de maneira permanente as instruções do programa), memória de dados (com a finalidade de memorizar os valores associados com as variáveis definidas no programa), uma série de pinos de entrada/saída (com a finalidade de realizar a comunicação do microcontrolador com o meio externo) e vários periféricos (tais como temporizadores, controladores de interrupção, temporizadores cão de guarda (WatchDog Timers – WDTs), comunicação serial, geradores de modulação por largura de pulso ou de PWM (Pulse Width Modulation), conversores analógico/digital etc.), fazendo com que o hardware final fique extremamente complexo. Isto é uma das características fundamentais que o diferencia dos microprocessadores, pois estes não possuem todos esses recursos em uma única pastilha. (MARTINS, 2005, p. 16)

Os microcontroladores podem ser considerados uma versão reduzida de um computador, uma vez que a função principal de ambos é processar informações.

O desempenho de um microcontrolador está diretamente ligado à qualidade e à capacidade de seus componentes. O microprocessador precisa interpretar de forma coesa as instruções fornecidas pelo programa que ele irá processar. Sua memória de programa deve conter espaço suficiente para armazenar as instruções

do programa, caso contrário, não será possível fazer o carregamento do programa para o microcontrolador.

De acordo com Monk (2013, p. 1) “Arduino é uma pequena placa de microcontrolador contendo um plugue de conexão USB(universal serial bus) que permite a ligação com um computador”. Por ser universal, esta comunicação através do USB é possível em praticamente todos os tipos modernos de computador. Isto permite uma gama maior de possibilidades a serem realizadas com o microcontrolador, não tendo a necessidade de se limitar a um nicho específico de máquinas e situações.

2.2 Arduino

O Arduino possui um microcontrolador que consegue realizar operações matemáticas e processar comandos oriundos de um programa. Uma qualidade do Arduino é que ele possui código aberto, além de vários códigos fonte como exemplos em sua IDE e na internet, o que permite a comunidade aprender mais sobre seu funcionamento e a criarem seus próprios códigos.

McRoberts explica de forma simplificada o que é e quais suas finalidades: “Em termos práticos, um Arduino é um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele.” (MCROBERTS, 2011, p. 22).

Segundo o próprio site¹ do Arduino: “Um Sketch é o nome que o Arduino usa para um programa. É a unidade de código que é carregada e executada em uma placa Arduino”. O Sketch é o programa utilizado na IDE do Arduino. Ele é o responsável pela definição de quais ações serão executadas em resposta das leituras realizadas.

Conforme Chavier(2018), no Sketch do Arduino a linguagem utilizada é a C++ porém com algumas pequenas alterações, uma linguagem de programação de alto nível muito conhecida. O fato dela ser bem conhecida é um possível fator determinante para a facilidade de busca por materiais que permitam seu aprendizado.

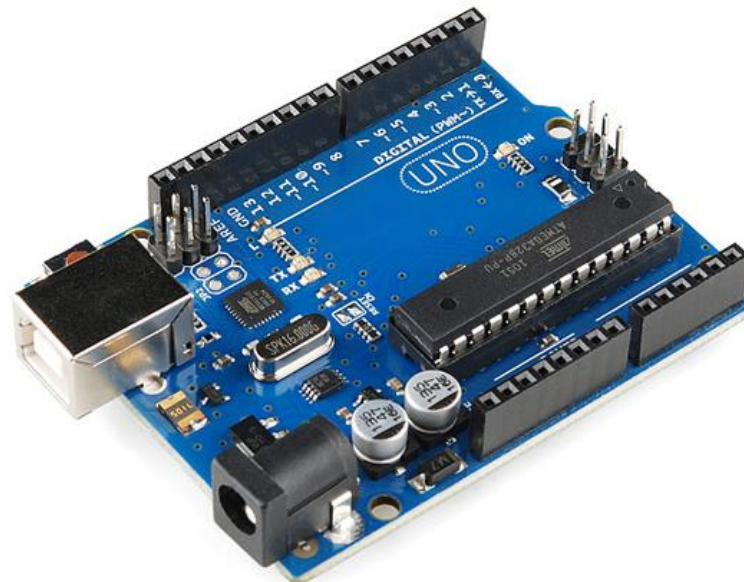
¹ <https://www.arduino.cc/en/tutorial/sketch>

2.2.1 Arduino Uno

O site do Arduino, na seção Boards lista vários modelos de placas Arduino como: Arduino Uno, Arduino Duemilanove, Arduino Diecimila, Arduino Mega, Arduino LilyPad, dentre diversos outros. Na página² onde especifica as características do modelo Uno diz que: “A UNO é a melhor placa para começar com eletrônica e codificação ... é a placa mais robusta com a qual você pode começar a brincar. A UNO é a placa mais utilizada e documentada de toda a família Arduino.”

A Figura 1 demonstra o modelo UNO que é muito bem recomendado para pessoas que estão iniciando seus projetos em Arduino. Outro importante fator é que diferente de outros modelos, que utilizam o chip FTDI, ele é integrado com um chip ATmega 328, que segundo MORGENSTERN; MACHADO e PADOIN (2015) possui um consumo energético menor em comparação com outros processadores como o Celeron430 da Intel. Isso contribui para que seu custo final seja menor.

Figura 1: Arduino Uno



Fonte: Filipeflop (2017)

² <https://store.arduino.cc/usa/arduino-uno-rev3>

2.3 Sensores

Em diversos trabalhos com uso de placas Arduino é possível notar a presença de sensores. Eles são os responsáveis pela percepção dos estímulos externos, durante a execução do sketch todas as informações de entrada serão recebidas através deles.

Atualmente, existem diversos tipos de sensores, criados exclusivamente para o Arduino, cada um com uma funcionalidade diferente, possibilitando assim a elaboração de diversos trabalhos com os mais distintos propósitos.

Conforme Patsko(2006, p.6) “O exemplo mais comum de sensor resistivo é o LDR (Light Dependent Resistor – Resistor Dependente de Luz)”, ele permite a percepção do nível de luz em um ambiente, muito úteis em projetos que precisem distinguir um ambiente claro e bem iluminado de um ambiente apresentando pouca ou a ausência de luz. Um grande exemplo deste recurso é encontrado em projetos voltados para a iluminação automática de locais quando estes estiverem escuros ou durante a noite.

Disponíveis ao Arduino estão vários outros sensores, como os de temperatura, toque, movimento, humidade dentre outros tipos, cada um com sua função.

Os sensores são divididos em dois grupos: os sensores digitais e os sensores analógicos.

2.3.1 Sensores Digitais

Patsko (2006) descreve sensores digitais como sendo sensores que captam níveis de tensão bem definidos, podendo assumir valores de “0” ou “1”, tensão alta (High) ou baixa (Low). Isto significa que, a tensão captada pode assumir apenas dois valores, não podendo haver um meio termo. As saídas digitais também assumem valores de tensão alta ou baixa para exibir seus resultados. Um exemplo de um sensor digital é o acionar de um botão, que do estado de baixa tensão passa diretamente para o estado de alta tensão.

2.3.2 Sensores Analógicos

Os sensores analógicos são diferentes dos sensores digitais por não se limitarem apenas a valores extremos de baixa ou alta tensão. Sensores analógicos são capazes de captar sinais analógicos. Conforme Patsko (2006, p. 1) “Sinais analógicos são aqueles que, mesmo limitados entre dois valores de tensão, podem assumir infinitos valores intermediários”. Um exemplo de sensor analógico é o sensor de temperatura, cuja temperatura registrada pode ser representada por vários valores, não se limitando apenas a tensão alta ou baixa.

2.4 Módulos para Arduino

Os módulos para Arduino são dispositivos que ao serem acoplados à placa, desempenham a função de entrada ou saída. Existe uma imensa diversidade de sensores para o uso no Arduino, alguns dos mais comuns são os sensores de luminosidade, temperatura, movimentos dentre outros, todos realizam uma leitura das ações ou condições sobre eles aplicadas e transmitem essa informação para o Arduino. Já os dispositivos de saída realizam uma ação dependendo da informação que lhes é fornecida, eles podem emitir luz, sons, movimentos, conforme o programa lhes permitir.

2.4.1 Joystick Analógico

O módulo joystick analógico é um sensor composto por 2 potenciômetros, cada um realizando a leitura analógica das medidas dos eixos X e Y, e um botão de pressão que recebe uma leitura digital do eixo Z.

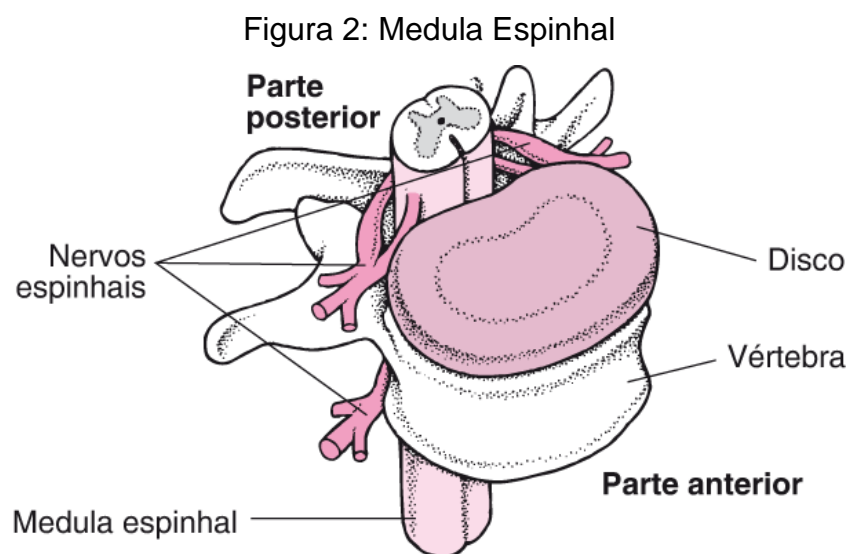
“Este é dotado de uma alavanca e composto internamente por dois potenciômetros de resistências: RX e RY, ambas de 10 k Ω que fazem entre si um ângulo de 90°, e um *push button* Z. Este dispositivo possui 5 terminais: +5V, GND, VRx, VRy e SW”(BRAGA et al. 2018).

2.5 Deficiência motora

Uma deficiência motora é caracterizada por algum tipo de inibição da capacidade motora da pessoa, podendo ela ser considerada parcial, causando a perda parcial dos movimentos dos músculos, como exemplo a tetraparesia, ou completa, causando a perda total do movimento voluntário dos músculos, como a paraplegia por exemplo.

2.6 Lesão medular

A medula espinhal, mostrada na Figura 2, é a parte do sistema nervoso central que se estende ao longo da coluna vertebral. Por se tratar de uma parte sensível do corpo, ela necessita de proteção, Gonçalves (2006) afirma que em volta da medula, como em todo o sistema nervoso central, encontra-se o líquido cefalorraquidiano, com a funcionalidade de protegê-lo, mas também susceptível a diversas doenças como a meningite e o surgimento de tumores.



Fonte: MSD (2018)

Gonzalez e Mattos descrevem em que consiste uma lesão medular:

Podemos dizer que lesão medular ou traumatismo raquimedular (TRM) é quando a medula espinhal é danificada seja por um trauma, doença ou defeito congênito, ocasionando paralisia temporária ou permanente dos músculos dos membros e do sistema nervoso autônomo, bem como alterações na sensibilidade dependendo da localização e extensão da lesão. (GONZALEZ e MATTOS. 2015).

Segundo a WHO³ (2013):

- Todo ano, ao redor do mundo, de 250.000 a 500.000 pessoas sofrem de algum tipo de lesão na medula espinhal.
- A maioria das lesões da medula espinhal acontecem por causas evitáveis, tais como falhas no trânsito, queda e violência.

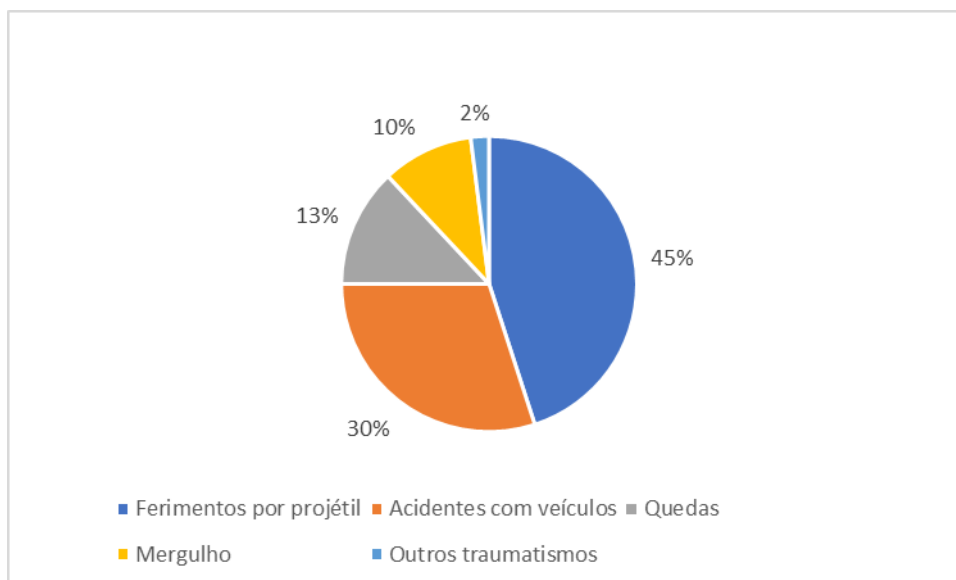
As principais causas das lesões na medula podem ser divididas em dois tipos com diferentes origens, as de origem traumática e as de origem não traumática.

2.6.1 Etiologia traumática

As lesões de origem traumática consistem no ferimento de forma direta da medula espinhal. Nas lesões traumáticas ocorre a fratura da coluna vertebral, seja por um forte impacto ou torção excessiva da mesma, e por se encontrar em volta da medula, a vértebra fraturada acaba lesionando a medula no local onde se localiza esta vértebra. Dentre os possíveis causadores de ferimentos, a Figura 3 mostra um gráfico em porcentagem das principais formas de lesão medular atualmente.

³ World Health Organization

Figura 3. Principais causas para traumas medulares.



Fonte. Gonzales e Mattos (2015).

2.6.2 Etiologia não traumática

Gonzales e Mattos definem quais os principais fatores das lesões não traumáticas

As causas não-traumáticas, são geradas por diversos fatores como: tumores que comprimem a medula espinhal ou as regiões próximas; acidentes vasculares; hérnia de disco; deformidades na coluna vertebral que afetam a integridade da medula espinhal. Dentre os causadores não traumáticos estão principalmente inflamações e problemas vasculares. (GONZALES E MATTOS, 2015).

2.7 Tetraplegia

Machado e Scramin (2005, p. 190) definem a tetraplegia como uma lesão medular completa no nível cervical, quanto mais alto for o nível desta lesão mais sensível será a situação desta pessoa. Acompanhados a esta lesão surgem diversas complicações na vida do indivíduo, como a redução da sensibilidade na região,

redução da força dos músculos, ocorre a alteração do equilíbrio, além do mal funcionamento de alguns órgãos internos do indivíduo. Fatores estes que comprometem a privacidade da pessoa por na maior parte do tempo necessitar de alguém o acompanhando e dando assistência, algo que pode gerar certo constrangimento, acompanhado da restrição das atividades de vida diária com autonomia.

3. METODOLOGIA

Os seguintes tópicos detalham quais procedimentos foram tomados para a realização deste trabalho.

3.1 Objeto de estudo

Todo o desenvolver deste trabalho, desde o início, realizando pesquisas, buscando em sites, livros e artigos, até a sua conclusão foram feitos com a intenção de ao final do projeto ter-se uma ferramenta prática, que possa ser útil no papel de substituir o mouse, para casos em que não haja a possibilidade do uso comum do mesmo. A criação da ferramenta destinando-se principalmente à pessoas que apresentam lesão na medula, bem como as que possuem diferentes tipos de distrofias musculares, ou quaisquer incapacidades motoras ou limitação dos membros superiores.

3.2 Componentes

Para a realização deste trabalho foi adquirido um Kit Arduino Start contendo vários componentes como: uma placa Arduino modelo UNO, um cabo USB, uma protoboard 400 pontos, jumpers Macho-Macho, resistores de 10K Ω , dentre outros que não foram usados para a elaboração deste projeto. Além deste Kit foi adquirido também um módulo joystick analógico para Arduino. A aquisição destes produtos teve o custo de 108 reais no site de vendas Eletrogate. A Tabela 1 apresenta os componentes físicos usados na ferramenta.

Tabela 1: Componentes Físicos

COMPONENTE	QUANTIDADE
Arduino	1
Protoboard	1
Cabo USB	1
Jumpers	7
Resistor 10KΩ	1
Joystick	1

Fonte: Próprio autor

Além deles foi necessário o uso da IDE do Arduino, para que fosse possível o upload do sketch que interpretaria as entradas recebidas e definir as ações correspondentes. Além desta IDE, foi utilizada a IDE do Netbeans para a compilação do código fonte responsável pelo movimento do cursor do mouse. A Tabela 2 demonstra estes programas necessários.

Tabela 2

PROGRAMA	FUNÇÃO DO CÓDIGO FONTE UTILIZADO
Arduino IDE	Receber entradas e exibir no Serial
Netbeans IDE	Ler dados do Serial e definir ações correspondentes

Fonte: Próprio autor

3.2.1 Montagem

Durante o processo de montagem foram consultados sites, vídeos e pessoas, a fim de ampliar o conhecimento acerca do Arduino e quais procedimentos deveriam

ser tomados para que este projeto fosse realizado.

A início foram realizados testes com a placa e o joystick a fim de verificar seu bom funcionamento e compreender melhor seu funcionamento.

O joystick foi acoplado à protoboard e seus conectores foram ligados ao Arduino, o GND ao GND, o +5V ao 5V, o VRX ao pino analógico A0, o VRY ao pino analógico A1 e o SW ao pino digital 7.

3.3 Sketch

Os programas utilizados, tanto o sketch quanto o código fonte em Java (que se encontram em anexo) foram disponibilizados em um site⁴ por um usuário chamado “Syntaxian”, que realizou um projeto parecido, também com o Arduino, em sua publicação ele auxilia outros usuários a fazer a comunicação entre o serial do Arduino e o mouse. Foram feitas apenas duas alterações do código original, uma corrigindo qual a porta usada pelo Arduino e outra modificando a velocidade do serial para que o controle do cursor seja mais fácil.

Pelo fato de o Arduino modelo Uno possuir o microcontrolador Atmega328, e este não possuir interface USB, não foi possível usar a biblioteca mouse.h, diferente do modelo Leonardo e outros, que possuem o microcontrolador Atmega 32U com interface USB.

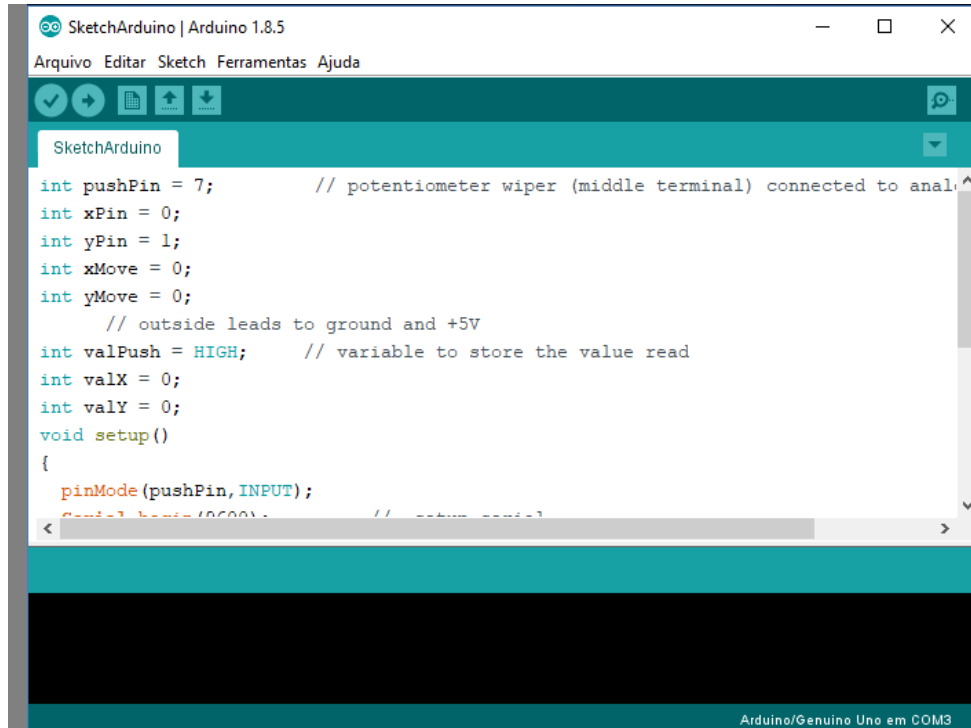
Como forma de contornar este empecilho, foi necessário o uso de API RxTx e da biblioteca JNA. A RxTx é responsável por possibilitar a leitura do serial do Arduino, já o JNA é uma biblioteca que possibilita de forma simples o acesso ao código nativo, que será utilizada para controlar a movimentação do cursor do mouse.

O Sketch representado na Figura 4 está programado para possibilitar a leitura analógica dos pinos 0 e 1, e realizar a leitura digital do pino 7. Cria as variáveis onde serão armazenados os valores dessa leitura e depois inicializa o serial a uma velocidade de 1200 bits por segundo. No loop ele vai exibindo no serial os valores atribuídos às variáveis que foram criadas para armazenar os valores das leituras

⁴<https://www.instructables.com/id/PC-Mouse-Made-With-Arduino-Uno-and-Joystick/>

analógicas e da digital.

Figura 4: Arduino IDE



Fonte: Próprio autor

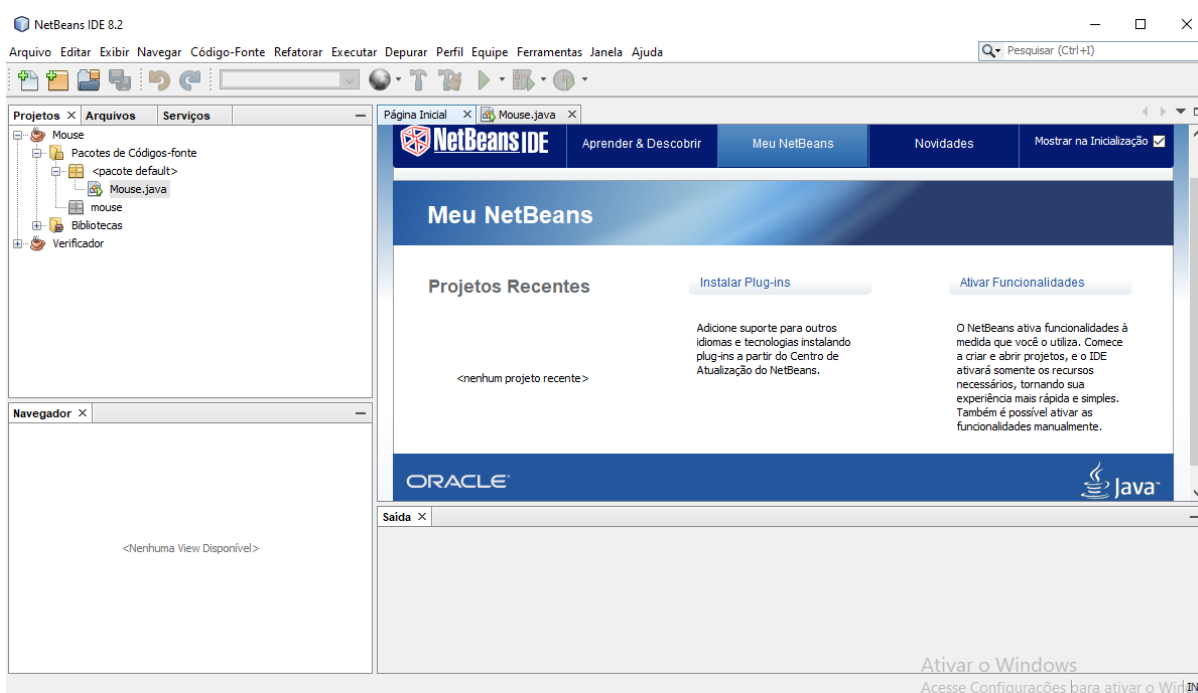
Para a utilização do RXTx e do JNA foi necessário baixá-los e transferir uma cópia dos arquivos: RXTXcomm.jar e jna-3.2.7-sources para o diretório ...\\Java\\jre1.8.0_191\\lib\\ext, e dos arquivos rxtxParallel.dll e rxtxSerial.dll para o diretório ...\\Java\\jre1.8.0_191\\bin.

Em seguida foi preciso criar um novo projeto em uma IDE com linguagem Java, a IDE escolhida foi o Netbeans, representada na Figura 5. Neste novo projeto foi adicionado um código fonte que identifica em qual porta da máquina que o Arduino está conectado, define a quantidade de bits por segundo e fica ouvindo-a, armazena os dados vindos dela em uma variável, que, ao ser lida, irão chamar a função mouseMove passando como parâmetros esses valores das variáveis. Na função main, o programa espera até que alguma leitura seja feita para que ele possa inicializar.

A função mouseMove separa os valores do potenciômetro X, os do potenciômetro Y e os do pushbutton do serial em 3 variáveis diferentes, dessa forma, cada variável fica responsável pela coordenada de um movimento ou do

clique do cursor.

Figura 5: Netbeans IDE



Fonte: Próprio Autor

3.4 DESENVOLVIMENTO DO SUPORTE

Para a elaboração do suporte foi necessário encontrar algum objeto que pudesse comportar o Arduino, a protoboard e ter uma passagem livre para o cabo USB e para o joystick. Depois de testar possíveis caixas para alocar o dispositivo, foi escolhida uma caixa porta-joias, pois seu volume era o suficiente para armazenar os componentes e, pelo fato dela ser de madeira, possuía resistência sem intensificar muito o peso. Nesta caixa foram realizadas medidas para que pudesse da melhor forma possível acomodar os componentes. Foi possível perceber que a altura da caixa estava ligeiramente maior do que era necessário, então foi feito um preenchimento em madeira no fundo, para que o Arduino permaneça elevado. A partir das medidas feitas, foram confeccionados dois orifícios, um no topo da caixa, por onde sairia o joystick, de forma que seu movimento não fosse comprometido, e o

outro foi feito na lateral, no local onde passaria o cabo USB. A Figura 6 mostra como ficou a caixa após estes procedimentos.

Figura 6: A caixa

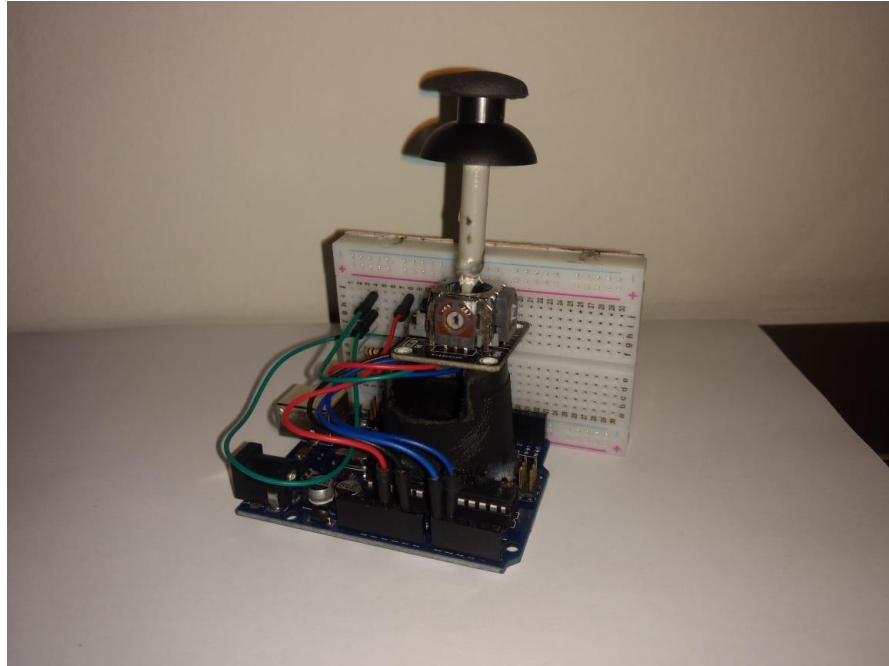


Fonte: Próprio autor.

Depois de pronta a caixa, foi identificado que, na posição em que o joystick estaria, seria difícil usar o botão, pois ele não se firmava apenas pelos conectores, necessitando de algum tipo de apoio para se escorar, que não tencionasse os jumpers. Ficou escolhido como material para a construção do apoio um cone de linha para costura vazio, pois este é rígido e poderia ser cortado de forma que não atrapalhasse os fios.

Além do joystick necessitar de um apoio, ele também precisou ser alongado, para que fosse possível o seu uso fora da caixa. Foi utilizado um cilindro de plástico em que suas pontas foram aquecidas e moldadas para se encaixarem ao joystick. A figura 7 representa como ficou o resultado destas ações.

Figura 7: Apoio e extensor



Fonte: Próprio autor

Vários foram os planos de possíveis suportes da Blackbox ao corpo. O suporte selecionado foi uma peça mostrada na Figura 8, geralmente usada para segurar o celular próximo ao rosto, ele é feito em um metal firme porém dobrável, possibilitando o ajuste dependendo do usuário.

Figura 8: Suporte



Fonte: Próprio autor

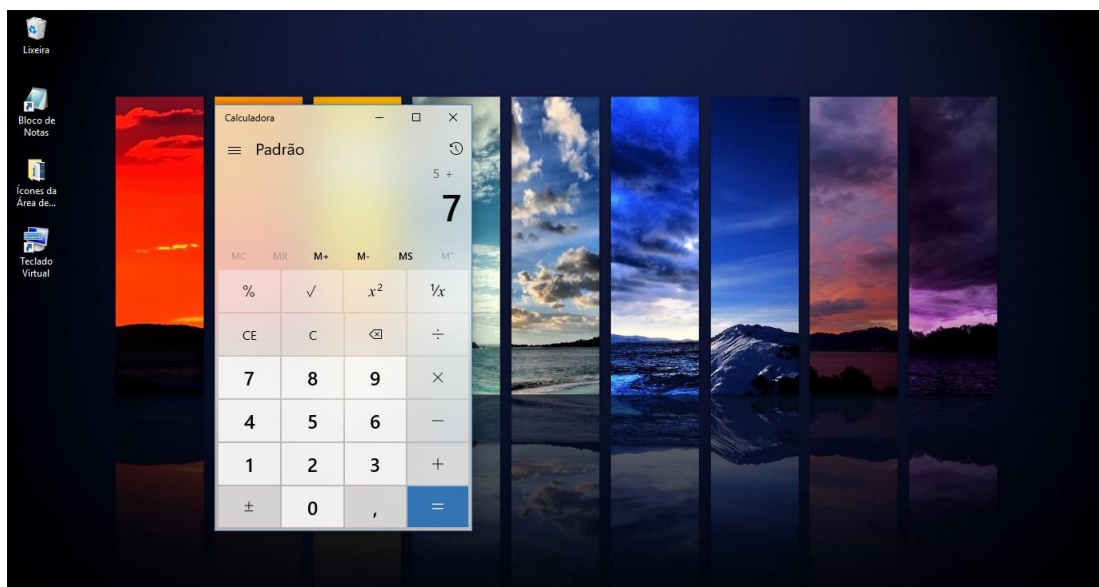
Várias peças foram compradas para a verificação da possibilidade do uso delas no suporte para a Blackbox, porém nem todas foram usadas. O preço somado das peças que foram usadas para a construção do suporte atinge a média de 60 reais.

3.5 REALIZAÇÃO DOS TESTES

Os testes executados para a avaliação do desempenho da Blackbox foram realizados pelo próprio autor. Estes testes foram realizados com a intenção de avaliar a capacidade de mobilidade do cursor, o tempo de reação entre o movimento do joystick e do cursor e o tempo gasto para a realização das atividades. Ao todo foram realizados dois testes.

O primeiro teste, representado na Figura X consistia em usar a calculadora virtual para realizar a conta: “5+7=?”, com o foco principal na avaliação da mobilidade do joystick, podendo ser identificada na precisão da seleção dos algarismos e símbolos, conforme mostrado na figura 9.

Figura 9: Teste da Calculadora

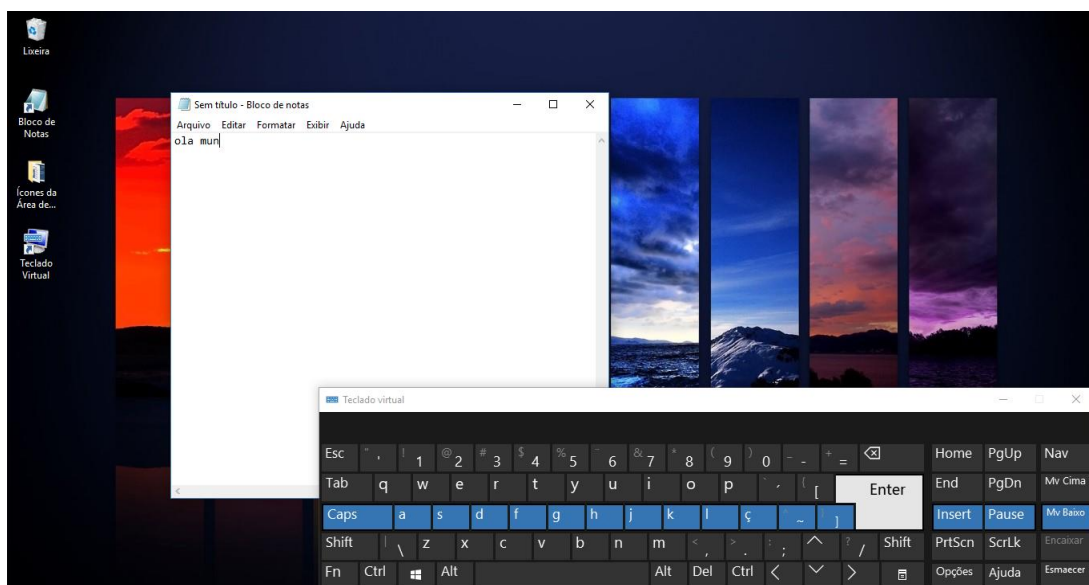


Fonte: Próprio autor

No segundo teste realizado, o usuário já possuía o teclado virtual à

disposição. A tarefa foi abrir o bloco de notas, nele, deveria digitar a frase “ola mundo” e salvar o documento no computador. Nesse teste foi avaliado principalmente o tempo de reação entre movimento do joystick e o tempo de reação do cursor. Ambos os testes foram cronometrados a fim de se obter uma média de tempo para a realização de cada ação.

Figura 10: Teste do texto



Fonte: Próprio autor

3.6 ANÁLISE DOS RESULTADOS

Durante a realização do primeiro teste já foi possível perceber que havia certo delay de aproximadamente 1 segundo entre o joystick e o cursor do mouse, este fator dificultou a realização das ações delegadas. Ao tentar pairar o cursor sobre um determinado número, eram necessários pequenos movimentos para correção do posicionamento do cursor. Por fim chegou-se ao final esperado da operação matemática, tento como tempo de realização 43 segundos.

Durante a realização do segundo teste foi aberto o bloco de notas com um duplo clique do joystick. Para digitar o texto no bloco de notas foi utilizado o recurso “Rastrear teclas” do teclado virtual, pois o modo tradicional não reconhecia o clicar do joystick sobre as teclas. O atraso entre a ação e a reação do cursor foi, no início, uma dificuldade aparente, causando erros na seleção das teclas, porém, depois de certo tempo, foi se tornando mais fácil compensar o atraso da reação do cursor. Ao

final foi salvo o arquivo no computador. Toda esta tarefa teve 3 minutos e 18 segundos como tempo de duração, ou seja, 198 segundos.

Tabela 3. Tempo de duração por teste

<i>Testes Realizados</i>	<i>Tempo de Duração</i>
<i>Teste 1</i>	43 segundos
<i>Teste 2</i>	198 segundos

Fonte: Próprio autor

3.7 DESAFIOS ENCONTRADOS

Durante o desenvolvimento deste projeto, vários desafios foram encontrados. A elaboração em prazo útil se mostrou uma complicação real, mesmo com o cronograma já elaborado.

O registro das etapas do projeto também teve certo grau de dificuldade, visto que cada detalhe é importante para o entendimento do processo como um todo, alguns detalhes acabam se perdendo.

O tempo de atraso entre a ação no joystick e o tempo de reação do cursor foi no início uma dificuldade considerável, não permitindo a precisão necessária para poder escrever de forma rápida, porém, com o tempo esta dificuldade foi se tornando mais fácil de se contornar, pois, por ser sempre o mesmo tempo de atraso, bastava apenas interromper o movimento pouco tempo antes do cursor chegar ao local desejado.

Acima das dificuldades anteriores, a maior foi conseguir possibilitar que o Arduino modelo Uno fosse usado para controlar o cursor do mouse, pelo fato do seu microcontrolador não possuir interface USB, impossibilitando o uso da principal biblioteca criada para esta função, sendo necessário usar de outras formas para que o controle do cursor fosse possível.

3.8 POSSÍVEIS APRIMORAMENTOS

A primeira modificação visível para este trabalho é a substituição do modelo da placa Arduino, alterando para uma de modelo Leonardo ou Due, pois estas possuem outros modelos de microcontrolador que possibilitam a interface USB, facilitando o processo do controle do cursor do mouse através do módulo joystick.

Outro possível aprimoramento seria a troca do módulo joystick por outro tipo de sensor de movimento mais completo e funcional, como uma câmera que seja capaz de interpretar movimentos como instruções, um exemplo deste caso é o Kinect possibilitando esta função no XBOX, onde os movimentos do jogador captados pelo Kinect se tornam instruções que comandam as ações do jogo.

É possível também a implementação de todo esse sistema sem a utilização de um cabo USB, realizando a comunicação através de bluetooth, integrando um módulo do Arduino criado especificamente para possibilitar este recurso.

Uma outra maneira de melhorar este dispositivo seria encontrar o que causa o atraso entre o joystick e o cursor, pois assim ele se tornaria mais preciso, rápido e eficiente.

3.9 TRABALHOS FUTUROS

Até o momento foram identificados dois possíveis trabalhos futuros. Um deles consiste na substituição do modelo Arduino utilizado, e o outro no aprimoramento da interface da ferramenta com o usuário.

Ao utilizar o modelo Leonardo por exemplo, por possuir o microcontrolador ATmega32u4, com interface USB, seria possível o uso da biblioteca mouse.h, me possibilitando utilizar um código fonte mais simples e prático, que possivelmente reduziria o tempo de resposta entre o sensor e o cursor do computador.

Neste trabalho foi utilizado um sensor de movimento estilo joystick, gerando dados a partir dos movimentos com ele realizados. Além dele outros dispositivos de entrada poderiam realizar esta captura dos comandos. Um exemplo é o sensor óptico, que possibilitaria a coleta de informações sem a necessidade da fixação do dispositivo ao usuário, podendo de moderada distância, realizar a captura dos comandos desejados pelo usuário.

4. CONCLUSÃO

Com este trabalho foi possível concluir que, a Blackbox de certa forma cumpre o seu objetivo proposto, ela é uma ferramenta de baixo custo, que possibilita o movimento do cursor do mouse sem a necessidade de ficar usando as mãos, porém isto não significa que ela já se encontre em sua versão final, ela ainda tem a possibilidade de ser aprimorada, seja através da adição de um módulo bluetooth, para que não haja a necessidade de um cabo ligando-a ao computador ou com a redução do tempo de delay entre o movimento do joystick e do cursor, que se mostrou um contratempo nas ações do usuário.

O Arduino nos oferece diversas possibilidades de projetos graças aos seus diversos modelos de placa. O Arduino é dinâmico, barato e possui muito material de suporte disponível na internet, tornando mais fácil o seu uso para iniciantes.

Os resultados obtidos através deste trabalho poderiam ter sido ampliados caso a escolha da placa Arduino tivesse sido pelo modelo Leonardo ou Due. A impossibilidade do uso da biblioteca mouse.h gerou a necessidade de soluções não tão efetivas. Acredita-se que o tempo de delay entre o joystick e o cursor do mouse possa ser reduzido caso ocorra a troca do modelo de Arduino selecionado.

Os microprocessadores apesar de possuírem um desempenho de processamento inferior quando comparados a processadores comuns, eles são bem úteis para projetos que não possam arcar com grandes orçamentos e que não necessitem de grande desempenho de processamento.

Em relação ao apoio às pessoas, cada atitude conta. Não importa a forma como é possível contribuir, nem o tamanho da ajuda, a partir do momento em que decidimos dedicar parte do nosso tempo em favor dos outros, estamos contribuindo para uma sociedade melhor. Apesar de a maioria das lesões medulares serem incuráveis, vale o esforço para tentar tornar a vida da pessoa mais agradável possível.

Através deste trabalho foi possível perceber a importância de facilitar o acesso à informação a todas as pessoas, pois essa é uma das possíveis formas de se reduzir as diferenças sociais e culturais existentes em nosso país.

A elaboração deste trabalho possibilitou o aprendizado e a prática de

programação em linguagem C++ e java, linguagens interessantes e com grandes possibilidades de projetos.

5. REFERÊNCIAS

ARDUINO. **Arduino**. Disponível em: <www.arduino.cc>. Acesso em: 21 mai, 2018.

ARDUINO, **Arduino Older Boards**. Disponível em: <https://www.arduino.cc/en/Main/Boards>. Acesso em: 27 nov 2018.

ARDUINO, **ARDUINO UNO REV3**. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>. Acesso em: 27 nov 2018.

ARDUINO, **Sketch**. Disponível em: <https://www.arduino.cc/en/tutorial/sketch>. Acesso em: 27 nov 2018.

BRAGA, Pedro Augusto da Silva, et al. **Desenvolvimento de um sistema de realidade aumentada para interação com crianças com transtorno do espectro autista**. 2018. Disponível em: <https://ssl4799.websitesequivo.com/swge5/PROCEEDINGS/PDF/CBA2018-1173.pdf>. Acesso em: 14 nov 2018.

CHAVIER, Luís Fernando. **Programação para Arduino – Primeiros Passos**. 2018. Disponível em: <https://www.circuitar.com.br/tutoriais/programacao-para-arduino-primeiros-passos/>. Acesso em 27 nov 2018.

GONÇALVES, Fabiana Santos. **Medula espinhal**. Disponível em: <https://www.infoescola.com/anatomia-humana/medula-espinhal/>. Acesso em 14 nov 2018.

GONZALEZ, N. ; MATTOS, S. **Lesão Medular**. 2015. Disponível em: http://www.novoser.org.br/espacao_informacao_lm.html. Acesso em 14 nov 2018.

MACHADO, Wiliam César Alves e SCRAMIN, Ana Paula. **Cuidado multidimensional para e com pessoas tetraplégicas: re-pensando o cuidar em enfermagem**. Ciência, Cuidado e Saúde. Maringá, v. 4, n. 2, p. 190.

MARTINS, N. A. **Sistemas Microcontrolados: Uma Abordagem com o Microcontrolador PIC 16F84.** São Paulo: Novatec, 2005. Disponível em: <<http://www.martinsfontespaulista.com.br/anexos/produtos/capitulos/203289.pdf>>. Acesso em 20 jun, 2018.

MICROBERTS, Michael, **Arduino Básico.** Tradução de Rafael Zanolli. São Paulo: Novatec Editora, 2011.

MONK, Simon. **Programação com Arduino >> Começando com Sketches.** Tradução de Antônio Laschuck. São Paulo: Editora Bookman, 2013.

MORGENSTERN, M. S.; MACHADO, L. R. F.; PADOIN, E. L. **Análise de desempenho de processadores ATmega328 utilizando Benchmark Linpack.** 2015. Disponível em: <http://www.inf.ufrgs.br/erad2015/downloads/p/ic/s1-5.pdf>. Acesso em: 27 nov 2018.

GOLDMAN, Steven A. **Medula Espinhal.** Manual MSD, Merck Sharp & Dohme. Disponível em: <https://www.msdmanuals.com/pt-br/casa/dist%C3%BAArbios-cerebrais,-da-medula-espinhal-e-dos-nervos/biologia-do-sistema-nervoso/medula-espinhal>. Acesso em 20 dez 2018.

PATSKO, Luís Fernando. **Tutorial aplicações, funcionamento e utilização de sensores.** 2006. Disponível em: <http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_aplicacoes_e_funcionamento_de_sensores.pdf>. Acesso em 20 jun, 2018.

PC Mouse Made With Arduino Uno and Joystick. Disponível em: <https://www.instructables.com/id/PC-Mouse-Made-With-Arduino-Uno-and-Joystick/>. Acesso em 17 nov 2018.

WHO, WORLD HEALTH ORGANIZATION. **Spinal Cord Injury**. Disponível em: <http://www.who.int/en/news-room/fact-sheets/detail/spinal-cord-injury>. Acesso em 17 nov 2018.

6. ANEXO A – CÓDIGOS FONTE UTILIZADOS

Sketch usado na IDE do Arduino

```

int pushPin = 7;    // potentiometer wiper (middle terminal) connected to analog pin 3
int xPin = 0;
int yPin = 1;
int xMove = 0;
int yMove = 0;
    // outside leads to ground and +5V
int valPush = HIGH; // variable to store the value read
int valX = 0;
int valY = 0;
void setup()
{
    pinMode(pushPin,INPUT);
    Serial.begin(1200);    // setup serial
    digitalWrite(pushPin,HIGH);
}

void loop()
{
    valX = analogRead(xPin); // read the x input pin
    valY = analogRead(yPin); // read the y input pin
    valPush = digitalRead(pushPin); // read the push button input pin

    Serial.println(String(valX) + " " + String(valY) + " " + valPush); //output to Java program
}

```

Código fonte usado no Netbeans

```

import java.awt.*;
import java.awt.event.InputEvent;
import java.io.BufferedReader;
import java.io.InputStreamReader;

```

```

import java.io.OutputStream;
import gnu.io.CommPortIdentifier;
import gnu.io.SerialPort;
import gnu.io.SerialPortEvent;
import gnu.io.SerialPortEventListener;
import java.util.Enumeration;

public class Mouse implements SerialPortEventListener {
    SerialPort serialPort;

    /** The port we're normally going to use. */
    private static final String PORT_NAMES[] = {
        "/dev/tty.usbserial-A9007UX1", // Mac OS X
        "/dev/ttyACM0", // Raspberry Pi
        "/dev/ttyUSB0", // Linux
        "COM3", // Windows***** (I changed)
    };
    /**
     * A BufferedReader which will be fed by a InputStreamReader
     * converting the bytes into characters
     * making the displayed results codepage independent
     */
    private BufferedReader input;
    /** The output stream to the port */
    private OutputStream output;
    /** Milliseconds to block while waiting for port open */
    private static final int TIME_OUT = 2000;
    /** Default bits per second for COM port. */
    private static final int DATA_RATE = 1200;

    int buttonOld = 1;

    public void initialize() {
        // the next line is for Raspberry Pi and
        // gets us into the while loop and was suggested here was suggested
http://www.raspberrypi.org/phpBB3/viewtopic.php?f...
        //System.setProperty("gnu.io.rxtx.SerialPorts", "/dev/ttyACM0"); I got rid of this
        CommPortIdentifier portId = null;
        Enumeration portEnum = CommPortIdentifier.getPortIdentifiers();
        //First, Find an instance of serial port as set in PORT_NAMES.

```



```

while (portEnum.hasMoreElements()) {
    CommPortIdentifier currPortId = (CommPortIdentifier) portEnum.nextElement();
    for (String portName : PORT_NAMES) {
        if (currPortId.getName().equals(portName)) {
            portId = currPortId;
            break;
        }
    }
}
if (portId == null) {
    System.out.println("Could not find COM port.");
    return;
}
try {
    // open serial port, and use class name for the appName.
    serialPort = (SerialPort) portId.open(this.getClass().getName(),
        TIME_OUT);
    // set port parameters
    serialPort.setSerialPortParams(DATA_RATE,
        SerialPort.DATABITS_8,
        SerialPort.STOPBITS_1,
        SerialPort.PARITY_NONE);
    // open the streams
    input = new BufferedReader(new InputStreamReader(serialPort.getInputStream()));
    output = serialPort.getOutputStream();
    // add event listeners
    serialPort.addEventListener(this);
    serialPort.notifyOnDataAvailable(true);
} catch (Exception e) {
    System.err.println(e.toString());
}
}
/**
 * This should be called when you stop using the port.
 * This will prevent port locking on platforms like Linux.
 */
public synchronized void close() {
    if (serialPort != null) {
        serialPort.removeEventListener();
        serialPort.close();
    }
}

```

```

    }
}
/**
 * Handle an event on the serial port. Read the data and print it. In this case, it calls the mouseMove
method.
 */
public synchronized void serialEvent(SerialPortEvent oEvent) {
    if (oEvent.getEventType() == SerialPortEvent.DATA_AVAILABLE) {
        try {
            String inputLine=input.readLine();
            mouseMove(inputLine);
            System.out.println("*****");
            //System.out.println(inputLine);
        } catch (Exception e) {
            System.err.println(e.toString());
        }
    }
}
// Ignore all the other eventTypes, but you should consider the other ones.
}
public static void main(String[] args) throws Exception    {
    Mouse main = new Mouse();
    main.initialize();
    Thread t=new Thread() {
        public void run() {
            //the following line will keep this app alive for 1000 seconds,
            //waiting for events to occur and responding to them (printing incoming messages to
console).
            try {Thread.sleep(1000000);} catch (InterruptedException ie) {}
        }
    };
    t.start();
    System.out.println("Started");
}

// My method mouseMove, takes in a string containing the three data points and operates the
mouse in turn
public void mouseMove(String data) throws AWTEException
{
    int index1 = data.indexOf(" ", 0);
    int index2 = data.indexOf(" ", index1+1);

```

```
int yCord = Integer.valueOf(data.substring(0, index1));
int xCord = Integer.valueOf(data.substring(index1 + 1 , index2));
int button = Integer.valueOf(data.substring(index2 + 1));
Robot robot = new Robot();

int mouseY = MouseInfo.getPointerInfo().getLocation().y;
int mouseX = MouseInfo.getPointerInfo().getLocation().x;

if (button == 0)
{
    if (buttonOld == 1)
    {
        robot.mousePress(InputEvent.BUTTON1_DOWN_MASK);
        robot.delay(10);
    }
}
else
{
    if (buttonOld == 0)
        robot.mouseRelease(InputEvent.BUTTON1_DOWN_MASK);
}

if (Math.abs(xCord - 500) > 5)
    mouseX = mouseX + (int)((500 - xCord) * 0.02);
if (Math.abs(yCord - 500) > 5)
    mouseY = mouseY - (int)((500 - yCord) * 0.02);

robot.mouseMove(mouseX, mouseY);

buttonOld = button;
System.out.println(xCord + ":" + yCord + ":" + button + ":" + mouseX + ":" + mouseY);
return;
}
}
```