

UNIOESTE – Universidade Estadual do Oeste do Paraná

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

**Desenvolvimento de um headmouse usando o
microcontrolador A-Star 32U4 Pro Micro**

Vitor De Angelis

CASCABEL

2017

VITOR DE ANGELIS

**DESENVOLVIMENTO DE UM HEADMOUSE USANDO O
MICROCONTROLADOR A-STAR 32U4 PRO MICRO**

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel.

Orientador: Prof. Dr. Adair Santa Catarina

CASCADEL

2017

VITOR DE ANGELIS

**DESENVOLVIMENTO DE UM HEADMOUSE USANDO O
MICROCONTROLADOR A-STAR 32U4 PRO MICRO**

Monografia apresentada como requisito parcial para obtenção do Título de *Bacharel em Ciência da Computação*,
pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos
professores:

Prof. Dr. Adair Santa Catarina (Orientador)

Colegiado de Ciência da Computação, UNIOESTE

Prof. Dr. Marcio Oyamada (Coorientador)

Colegiado de Ciência da Computação, UNIOESTE

Prof. Dr. Anibal Mantovani Diniz

Colegiado de Ciência da Computação, UNIOESTE

Prof. M.Sc. Edmar André Bellorini

Colegiado de Ciência da Computação, UNIOESTE

Cascavel, 23 de Agosto de 2017.

Lista de Figuras

Figura 1.1: Interface do software MouseKey	2
Figura 1.2: Projeto original desenvolvido por Martin Millmore	2
Figura 2.1: Linha do tempo das Tecnologias assistivas desde 1968	5
Figura 2.2: O dispositivo GlassOuse	7
Figura 2.3: O sensor de mordida acoplado ao GlassOuse	7
Figura 2.4: HeadMouse Nano®	8
Figura 2.5: Adesivo de Referência	8
Figura 2.6: QuadMouse	9
Figura 2.7: Microcontrolador Arduino A-Star 32U4 Micro	9
Figura 2.8: Sensor LSM9DS0	10
Figura 2.9: Conversor lógico	10
Figura 2.10: Esboço da Comunicação Microprocessador-Sensor	11
Figura 2.11 Fluxograma do código Arduino	12
Tabela 2.1 - Tabela de preços dos componentes eletrônicos utilizados	13
Figura 3.1: Esquemático logic level converter	15
Figura 3.2: Divisão de canais do conversor	16
Figura 3.3: Divisor de tensão	17
Figura 3.4: Esquema do conversor de tensão	17
Figura 3.5 Representação visual dos três sensores (acelerômetro, giroscópio e magnetômetro)	18
Figura 3.6: Sensor LSM9DS0	18
Figura 3.7: Três jumpers do LSM9DS0 no dorso do componente	19
Figura 3.8: Esquemático dos Jumpers do LSM9DS0	19
Figura 3.9: Pinagem do Arduino Pro Micro	20
Figura 4.1: Projeto do primeiro protótipo do Headmouse	23
Figura 4.2: Protótipo final do Headmouse	23
Figura 4.3: Fixação do protótipo na cabeça	24
Figura 4.4: Sensor de mordida	24
Figura 4.5: Uso do sensor de mordida	25
Figura 4.6: Diagrama de Máquina de Estados do Clique do Mouse	25
Figura 4.12: Método inicialize	27
Figura 4.13: Aplicação Java para comunicação Serial	28
Figura 4.14: Tela da letra “b” do teclado virtual MouseKey	29
Figura A.1 “Installing device driver software”	33

Figura A.2 “Device driver software was not successfully installed”	33
Figura A.3 Prompt de comando	34
Figura A.4 Update Driver Software	34
Figura A.5 “Update Driver Software”	35
Figura A.6 Nova entrada do DriverManager	35
Figura A.7 Additional Boards Manager URLs	36
Figura A.8 Tools/Board/Boards Manager	36
Figura A.9 SparkFun AVR Boards by SparkFun Electronics	36
Figura A.10 Produtos Arduinos SparkFun	37
Figura B.1: Implementação do estado 0	38
Figura B.2: Implementação dos estados 1, 3 e 5	38
Figura B.3: Implementação dos estados 2 e 4	39
Figura B.4: Implementação do estado 6	39
Figura B.5: Implementação do estado 100	40

Lista de Abreviaturas e Siglas

CEPE	Conselho de Ensino Pesquisa e Extensão
CSG	<i>Chip Select Gyro</i>
CSXM	Seleciona entre SPI e I ² C
DRDYG	Dados do giroscópio prontos
GND	<i>Ground</i>
HV	<i>High voltage</i>
I ² C	<i>Inter-Integrated Circuit</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	<i>Integrated Development Environment</i>
INT1XM	Interrupção 1 do Acelerômetro e Magnetômetro
INT2XM	Interrupção 2 do Acelerômetro e Magnetômetro
LSC	Laboratório de Sistemas Computacionais
LV	<i>Low voltage</i>
PEE	Programa de Educação Especial
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
SDOG	Quando SPI: Saída de dados do giroscópio Quando I ² C: endereço I ² C
SDOX	Quando SPI: Saída de dados do giroscópio Quando I ² C: endereço I ² C
SPI	<i>Serial Peripheral Interface</i>
TA	Tecnologia Assistiva
TWI	<i>Two Wires Interface</i>
UNIOESTE	Universidade Estadual do Oeste do Paraná
USB	<i>Universal Serial Bus</i>
VCC	Rótulo de um pino de fonte de alimentação
VDD	Rótulo de um pino de fonte de alimentação

Lista de Símbolos

Ω	Unidade de medida de resistência elétrica, Ohm
Gs	Unidade de medida de indução magnética, Gauss
Hz	Unidade de medida de frequência, Hertz
m	Unidade de medida de comprimento, metros
s	Unidade de medida de tempo, segundos
V	Unidade de medida de tensão elétrica, Volts

Sumário

LISTA DE FIGURAS	III
LISTA DE ABREVIATURAS E SIGLAS	V
LISTA DE SÍMBOLOS	VI
SUMÁRIO	VII
RESUMO	IX
CAPÍTULO 1	1
INTRODUÇÃO	1
1.1 Objetivos	2
1.2 Organização do Trabalho	3
CAPÍTULO 2	4
REVISÃO DE LITERATURA	4
2.1 Histórico da Tecnologia Assistiva	4
2.2 Tecnologia Assistiva no Brasil	5
2.3 Alguns Exemplos de Tecnologias Assistivas	6
2.4 Projeto de Headmouse desenvolvido por Millmore	9
CAPÍTULO 3	14
MATERIAIS E MÉTODOS	14
3.1 Componentes eletrônicos	15
3.1.1 Conversor lógico bidirecional.....	15
3.1.2 Divisor de tensão	16
3.1.3 LSM9DS0.....	18
3.1.4 Pro Micro Atmega32U4.....	20
CAPÍTULO 4	22
RESULTADOS E DISCUSSÃO	22
4.1 O protótipo do Headmouse	22
4.1.1 Divisor de tensão	24
4.1.2 Sensor de mordida	24
4.1.3 Software cliente	26
4.2 Testes Realizados	28
CAPÍTULO 5	31

CONSIDERAÇÕES FINAIS.....	31
5.1 Trabalhos Futuros.....	32
ANEXO A.....	33
A.1 Preparação do Ambiente Windows/Arduino Pro Micro.....	33
A.1.1 Instalação do <i>Driver</i> Sparkfun Pro Micro	33
A.1.2 Adicionando o Sparkfun Pro Micro à Arduino IDE	35
ANEXO B.....	38
B.1 Código da máquina de estados que implementa o clique do mouse com sensor de mordida.....	38
REFERÊNCIAS	41

Resumo

Neste trabalho desenvolveu-se um *headmouse* para que pessoas, com mobilidade dos braços reduzida, possam comandar o cursor do mouse utilizando movimentos da cabeça. O *headmouse* desenvolvido baseou-se no projeto de Martin Millmore e teve suas funcionalidades expandidas. O usuário pode controlar os movimentos do cursor, quando no modo “leitura”, e efetuar cliques simples e duplos com os botões do mouse, quando no modo “interação”. Para tetraplégicos que controlam sua mordida, incorporou-se ao projeto um sensor que emula os cliques do mouse de acordo com o número de mordidas efetuadas. Em testes realizados com paciente do Centro de Reabilitação Física da UNIOESTE, percebeu-se que o *headmouse* atendeu aos fins para o qual foi desenvolvido. As novas funcionalidade incorporadas ao *headmouse* permitiram-na digitar textos simples, sem o auxílio de terceiros, o que não era possível até este momento. Assim, ampliaram-se os limites de utilização do computador para esta paciente, que pode agora utilizar softwares, acessar a Internet e empregar os meios de comunicação eletrônicos, obtendo melhor qualidade de vida.

Palavras-chave: headmouse, arduino, giroscópio, deficiência motora.

Capítulo 1

Introdução

Segundo dados do IBGE [1], obtidos pelo censo demográfico realizado em 2010, aproximadamente 7% da população brasileira, próximo de 13,2 milhões de pessoas, possui restrições nos movimentos do corpo em algum nível, entre absoluto e moderado. Devido à restrição de movimentos nos membros, principalmente superiores, essas pessoas têm dificuldades para utilização de computadores, pois as interfaces gráficas são manipuladas por dispositivos apontadores, como o mouse.

No ano de 2016, o Programa Institucional de Ações relativas às Pessoas com Necessidades Especiais (PEE) procurou professores do Curso de Ciência da Computação e apresentou uma dificuldade enfrentada por uma discente com tetraplegia. Esta discente utiliza o computador como mecanismo auxiliar no processo de comunicação, principalmente na redação de textos por meio do MouseKey [2], ilustrado na Figura 1.1. O MouseKey é um teclado virtual desenvolvido por Cláudio Luciano Dusik e distribuído gratuitamente pela Universidade Federal do Rio Grande do Sul. Consiste em um teclado Silábico-Alfabético, que possui letras e padrões silábicos em torno de cada letra. O usuário seleciona as famílias de letras para formar palavras, o que diminui o número de movimentos e esforços necessários para escrever.

A discente em questão utiliza como dispositivo apontador, que interage com o teclado virtual MouseKey, o software Headmouse [3]. Este software foi desenvolvido pela Universidade de Lleida (UdL), na Espanha, com o objetivo de permitir o controle do mouse através do movimento da cabeça. Ele facilita o uso dos computadores, empregando conceitos de visão artificial, convertendo os movimentos da cabeça do usuário, capturados por uma *webcam*, em movimentos do cursor na tela. Entretanto, o software Headmouse apresenta falta de precisão no movimento do cursor, há necessidade de recalibragem após algum tempo de uso e seu uso prolongado causa-lhe fadiga.



Figura 1.1: Interface do software MouseKey

Fonte: NIEE UFRGS [4]

Motivados pelos problemas apresentados pelo PEE, professores do Curso de Ciência da Computação iniciaram pesquisas, através da Internet, para possíveis soluções, encontrando o projeto de um headmouse embarcado, desenvolvido por Martin Millmore [5]. O projeto pode ser visualizado na Figura 1.2.

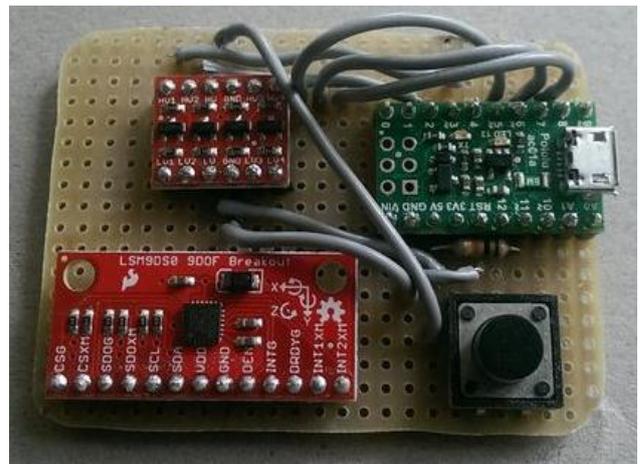


Figura 1.2: Projeto original desenvolvido por Martin Millmore

Fonte: Millmore [5]

1.1 Objetivos

Este trabalho tem o objetivo de construir e adaptar o *headmouse*, desenvolvido por Millmore, para que pessoas com mobilidade dos braços reduzida possam comandar o cursor do mouse utilizando movimentos da cabeça. Espera-se que esta tecnologia facilite o uso de computadores, possibilite o acesso a informações através da Internet e o emprego de meios de

comunicação eletrônicos, proporcionando melhor qualidade de vida para pessoas com alguma deficiência motora.

Como objetivos específicos:

1. Construir o primeiro protótipo empregando o projeto de Martin Millmore;
2. Integrar o hardware construído com o software de controle;
3. Realizar primeiro conjunto de testes;
4. Propor alterações no projeto, como o sensor de mordida (clique) e comunicação com o software cliente no computador para alternar entre modo leitura ou interação;
5. Realizar testes após os ajustes propostos.
6. Retornar ao passo 4 se necessário, até que o dispositivo seja considerado utilizável.

1.2 Organização do Trabalho

A seguir, tem-se um roteiro que engloba a organização do trabalho e uma breve descrição de cada capítulo:

- *Capítulo 1 - Introdução:* uma apresentação rápida, que transmite ao leitor a abrangência, a justificativa e os objetivos do trabalho.
- *Capítulo 2 – Revisão de Literatura:* apresenta uma perspectiva histórica da tecnologia assistiva, algumas das tecnologias assistivas e o projeto de Martin Millmore.
- *Capítulo 3 – Materiais e Métodos:* descreve os componentes usados no protótipo.
- *Capítulo 4 – Resultados e Discussão:* mostra a fase final do projeto, modificações realizadas e testes.
- *Capítulo 5 – Considerações Finais:* conclusões acerca das funcionalidades propostas e recomendações para trabalhos futuros.

Capítulo 2

Revisão de Literatura

Neste capítulo apresenta-se o histórico das Tecnologias Assistivas (TA) e como as iniciativas e legislação brasileiras têm contribuído para o desenvolvimento da área em nosso país. Também são apresentados alguns dispositivos comerciais destinados a atender pessoas com restrições de movimentos. Por fim, apresentamos, brevemente, informações relativas ao projeto de *headmouse* desenvolvido por Millmore.

2.1 Histórico da Tecnologia Assistiva

A TA sempre esteve presente na história da humanidade, desde o uso de uma simples bengala até a elaboração de sistemas para substituição de funções primariamente motoras. Segundo definição da *Assistive Technology Industry Association* [6], “tecnologia assistiva é qualquer item, peça de equipamento, software ou produto que é usado para aumentar, manter ou melhorar a capacidade funcional de pessoas com deficiência”.

De acordo com o conceito apresentado, as TA podem ter sido utilizadas desde o período a.C.. Entretanto, a eletrônica associada ao desenvolvimento de computadores e periféricos correspondentes tem iniciativas mais recentes. Com o avanço da tecnologia, a TA ganha novos e melhores recursos para se desenvolver, possibilitando a construção de uma sociedade mais inclusiva.

Na Figura 2.1 vemos uma linha do tempo com as principais inovações da TA desde 1968, englobando diversos tipos de tecnologia para os mais variados tipos de deficiência.

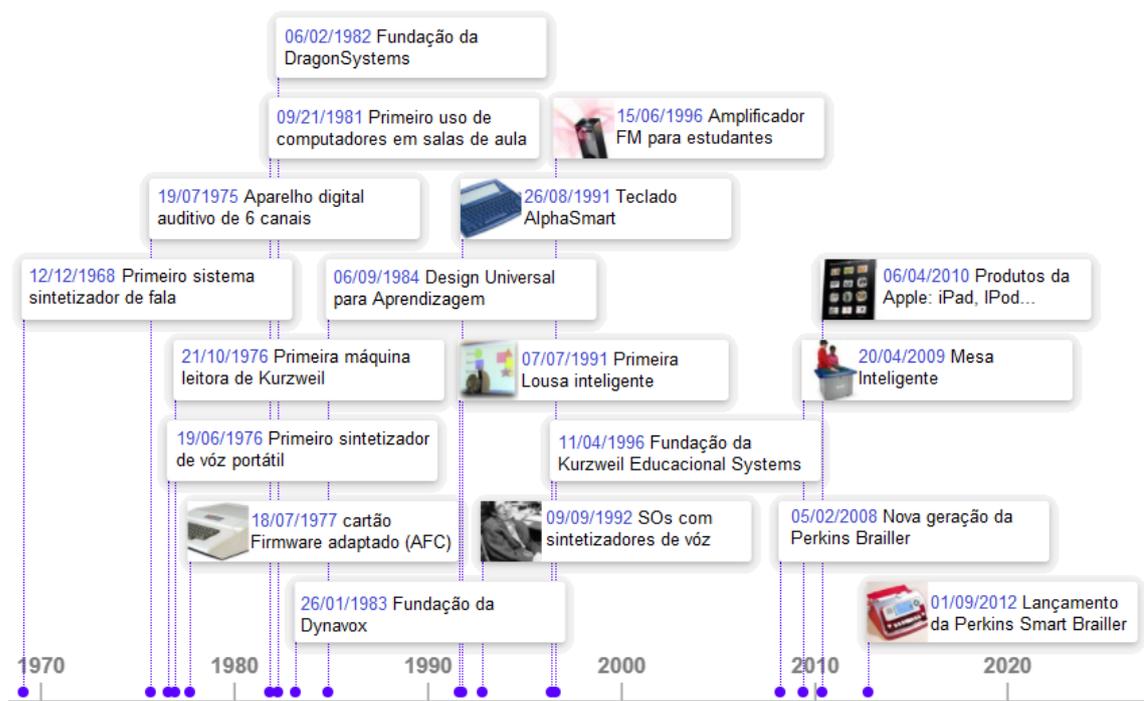


Figura 2.1: Linha do tempo das Tecnologias assistivas desde 1968

Fonte: Adaptado de History of Assistive Technology [7]

2.2 Tecnologia Assistiva no Brasil

As pessoas com algum tipo de deficiência fazem parte de 23,9% da população Brasileira, segundo os dados do Censo 2010, realizado pelo IBGE [8]. Em termos de deficiência motora são aproximadamente 7% da população.

Para oferecer melhores condições de vida à parcela tão expressiva da população, a área de TA encontra-se em crescimento nos últimos anos, como consequência de alguns fatores que impulsionam demandas de recursos e serviços destinados às pessoas com deficiência. O principal desses fatores refere-se ao destaque que se tem dado aos arranjos sociais como promotores de acessibilidade para essas pessoas. Nessa concepção são questionados todos os mecanismos que, de alguma forma, impedem a participação plena dessa parcela da população nos diferentes espaços e papéis sociais. Para tanto, buscam-se formas de garantir efetivamente tal participação como direito de todos [9].

Atualmente, existem no Brasil políticas públicas que contribuem para aumentar a demanda das TA em larga escala. Uma dessas políticas é a Política nacional de Educação Especial na Perspectiva Inclusiva [10], que consiste numa ação política, cultural, social e pedagógica, desencadeada em defesa do direito de todos os alunos à inclusão, a fins de aprendizado e

participação, sem nenhum tipo de discriminação. A TA pode contribuir para a integração desses alunos, não somente no ambiente de ensino, como também no âmbito social.

Outra contribuição importante são as Políticas de Inclusão Digital [11], que propõem a disseminação da Internet em todo o Brasil, por meio de telecentros, e a capacitação da comunidade que auxilia na inclusão digital das pessoas. Mesmo que tal política não faça referência direta à necessidade de recursos de TA nesses telecentros, podemos presumir que isso é essencial, diante da diversidade do público ao qual se destina, e também em decorrência das leis brasileiras que garantem a acessibilidade em diversos espaços [9].

É claro que não existem apenas essas políticas sociais para incentivo do desenvolvimento da TA no Brasil. Pode-se citar, como exemplo, o Decreto 5.296/04 [12] que torna obrigatório que os portais e sítios eletrônicos da administração públicas garantam a acessibilidade para pessoas com deficiência visual, e o Plano Nacional dos Direitos da Pessoa com Deficiência (programa Viver sem Limite) [13], que visa implementar novas iniciativas e intensificar ações desenvolvidas pelo governo em benefício das pessoas com deficiência.

O Programa Institucional de Ações Relativas às Pessoas com Necessidades Especiais (PEE) [14] da Universidade Estadual do Oeste do Paraná - UNIOESTE, constituído pela Resolução nº 323/97-CEPE e regulamentado pela Resolução nº 319/2005-CEPE, se insere num conjunto de conquistas obtidas pela comunidade organizada das pessoas com deficiência, tendo como objetivo principal contribuir para o processo do saber sistematizado dos alunos pertencentes a esta comunidade.

Mesmo com todos os incentivos que existem, não significa que exista a garantia de plena igualdade de oportunidades para todos. É corriqueira a adequação das pessoas com deficiência à falta de acessibilidade, o que dificulta sua completa participação em atividades sociais.

2.3 Alguns Exemplos de Tecnologias Assistivas

Empresas como a GlassOuse [15], Origin Instruments [16] e Broadened Horizons [17], visando atender às necessidades do público abordado neste trabalho, desenvolveram TA com controles específicos, que resultam na mesma função de aparelhos convencionais, como um mouse comum.

O dispositivo GlassOuse (Figura 2.2), semelhante a um óculos, permite o controle de diversos equipamentos eletrônicos sem o uso das mãos. Um usuário do GlassOuse realiza apenas movimentos da cabeça para posicionamento do cursor, e um sensor de mordida para o

clique do mouse (Figura 2.3). Os óculos podem se conectar a outros dispositivos por meio de *Bluetooth*[®] e têm uma bateria com duração de aproximadamente sete dias. Podem ser importados por US\$399,00 [15].



Figura 2.2: O dispositivo GlassOuse
Fonte: GlassOuse Assistive Device [15]



Figura 2.3: O sensor de mordida acoplado ao GlassOuse
Fonte: GlassOuse Assistive Device [15]

Um segundo exemplo de TA é o HeadMouse[®] Nano [16], apresentado na Figura 2.4. Este dispositivo dispõe de um sensor óptico sem fio e um adesivo de referência, que deve ser posicionado na testa do usuário (Figura 2.5). Tal mecanismo permite o controle do cursor do mouse apenas com movimentos da cabeça e o clique através de um piscar de olhos. Pode ser importado a partir de US\$995,00¹ [18].

¹ Price List, Part Number HM-0220



Figura 2.4: HeadMouse Nano®

Fonte: Origin Instruments [16]



Figura 2.5: Adesivo de Referência

Fonte: Origin Instruments [16]

Outro exemplo de TA disponível é o QuadMouse [17], ilustrado na Figura 2.6, que pode ser controlado por bochechas, lábios ou língua. O aparelho é conectado ao computador por cabo USB e pode ser fixado com um suporte em diversas superfícies. Pode ser adquirido por US\$700,00 [17], em importação.



Figura 2.6: QuadMouse

Fonte: Broadened Horizons [17]

2.4 Projeto de Headmouse desenvolvido por Millmore

O projeto de *headmouse* desenvolvido por Millmore emprega um microcontrolador Arduino. Arduino [19] é uma plataforma de hardware e software de código aberto e de fácil uso. As placas Arduino são capazes de ler entradas, como sensor de luz, botões de pressão, entre outros, e transformá-las em saídas como acender um LED ou ativar um motor. A placa é controlada por instruções enviadas ao microcontrolador integrado. Para tal feito é usada a linguagem de programação Arduino e o ambiente de desenvolvimento integrado (IDE) Arduino Software. O projeto emprega também um giroscópio, que é capaz de fornecer informações relativas ao movimento no espaço 3D.

O projeto original usa uma placa Arduino A-Star 32U4 Micro (Figura 2.7), clone do Arduino Leonardo, operando a 5 V em 16MHz, com 32 Kbytes de memória *flash*, sendo 4 Kbytes reservados para a *bootloader* (um trecho de código que é executado quando a placa é ligada).

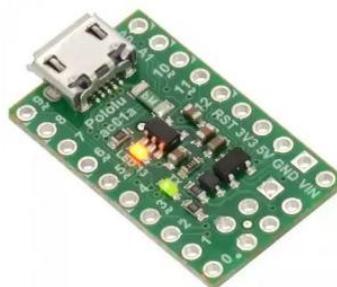


Figura 2.7: Microcontrolador Arduino A-Star 32U4 Micro

Fonte: POLULU [20]

O sensor LSM9DS0 [21] (Figura 2.8) é o mesmo do projeto original. O chip é a combinação de um giroscópio, acelerômetro e magnetômetro, fazendo com que a detecção de movimento seja sofisticada, levando em consideração que há possibilidade de medir velocidade angular, aceleração do componente e detectar a orientação de campos eletromagnéticos. O LSM9DS0 funciona com saídas de 3.3 V, porém, o A-Star 32U4 Micro precisa de entradas de 5 V. Então, torna-se necessário usar um conversor lógico (Figura 2.9), interligando os dois componentes para garantir a comunicação das linhas de dados: Serial Data (SDA) e Serial Clock (SCL).

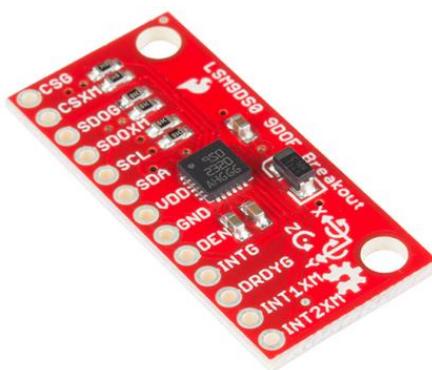


Figura 2.8: Sensor LSM9DS0

Fonte: SparkFun [22]

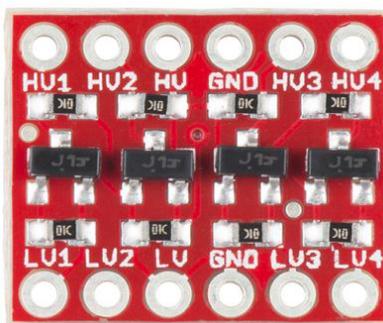


Figura 2.9: Conversor lógico

Fonte: SparkFun [23]

O melhor modo de utilizar o LSM9DS0 é por meio de uma interface que utiliza o protocolo de comunicação I^2C (*Inter-Integrated Circuit*).

I^2C é um protocolo de interface de dois fios (TWI – *Two Wires Interface*), desenvolvido pela NXP Semiconductors [24], que conecta dispositivos de baixa velocidade como microcontroladores, interfaces de comunicação e periféricos em sistemas embarcados [25]. Tornou-se popular por ser de uso simples; com apenas dois fios é possível fazer a conexão de

um número ilimitado de dispositivos com comunicação I²C.

A comunicação I²C permite que o microcontrolador se comunique com o giroscópio, por uma interface serial simples, a qual é suportada pelas bibliotecas Arduino padrão. Para isso acontecer, são conectadas as portas SDA e SCL do giroscópio às portas digitais 2 (SDA) e 3 (SCL) do Arduino A-Star 32u4, via conversor lógico, conforme ilustrado na Figura 2.10.

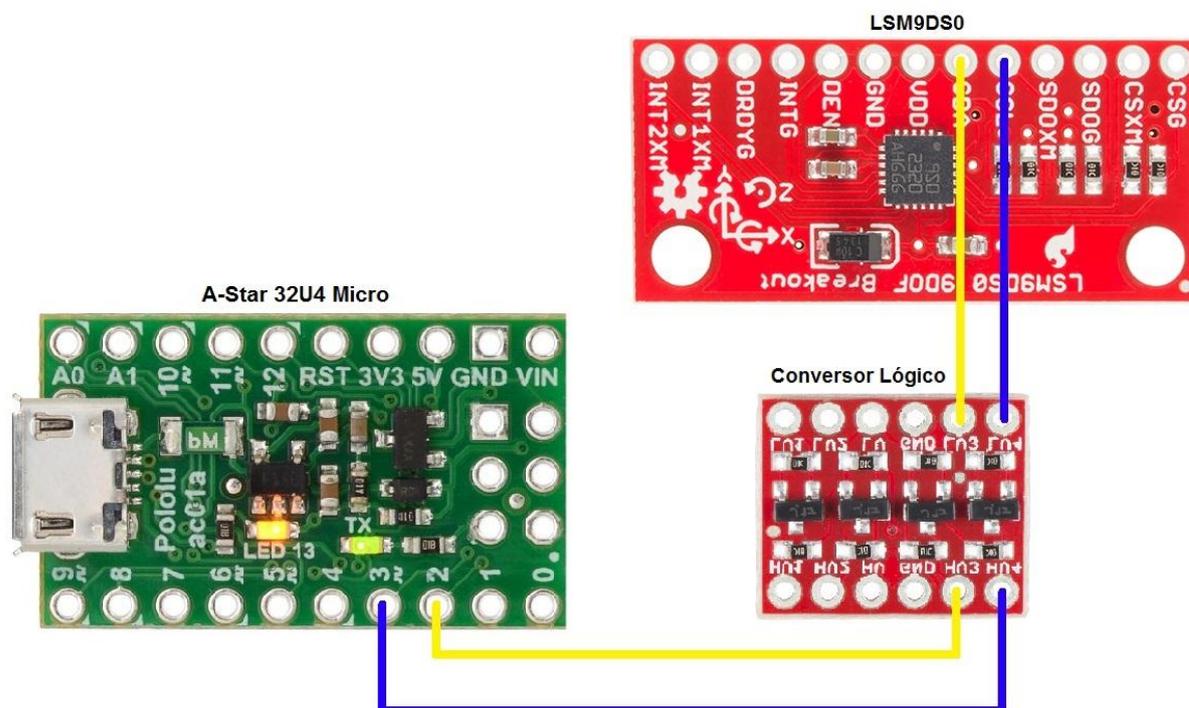


Figura 2.10: Esboço da Comunicação Microprocessador-Sensor

No desenvolvimento do projeto, Martin detectou uma falha no LSM9DS0. Os dados retornados pelo sensor nem sempre são os esperados. Para isso, foram encontradas bibliotecas que corrigem os problemas no SparkFun Education [26], comunidade que abriga muitos recursos sobre circuitos eletrônicos.

O autor combina código obtido no Sparkfun Education com as leituras do acelerômetro, para detectar como o dispositivo está orientado e transladá-lo nos eixos x e y correspondentes ao movimento do mouse. O dispositivo deve ser usado do lado direito da cabeça, caso contrário, o cursor do mouse se moverá no sentido inverso aos movimentos da cabeça.

O LSM9DS0 tem uma inicialização mais lenta do que o microcontrolador, então o código força uma pausa de 1 segundo quando inicializa os sensores. As 10 primeiras leituras do sensor são descartadas por não serem precisas.

Os dados fornecidos pelo acelerômetro não são nulos quando o equipamento está parado,

então o código faz uma média da décima até a quadragésima leitura, subtraindo-a dos demais valores lidos para garantir a estabilidade. Por conta disso, o dispositivo deve estar estável quando ligado. O botão é usado para pausar a interação com o mouse e, quando a interação é retomada, as leituras de dados são refeitas e as médias recalculadas; nesse momento o dispositivo também deve estar imóvel.

A Figura 2.11 apresenta um fluxograma que demonstra o funcionamento geral do código Arduino.

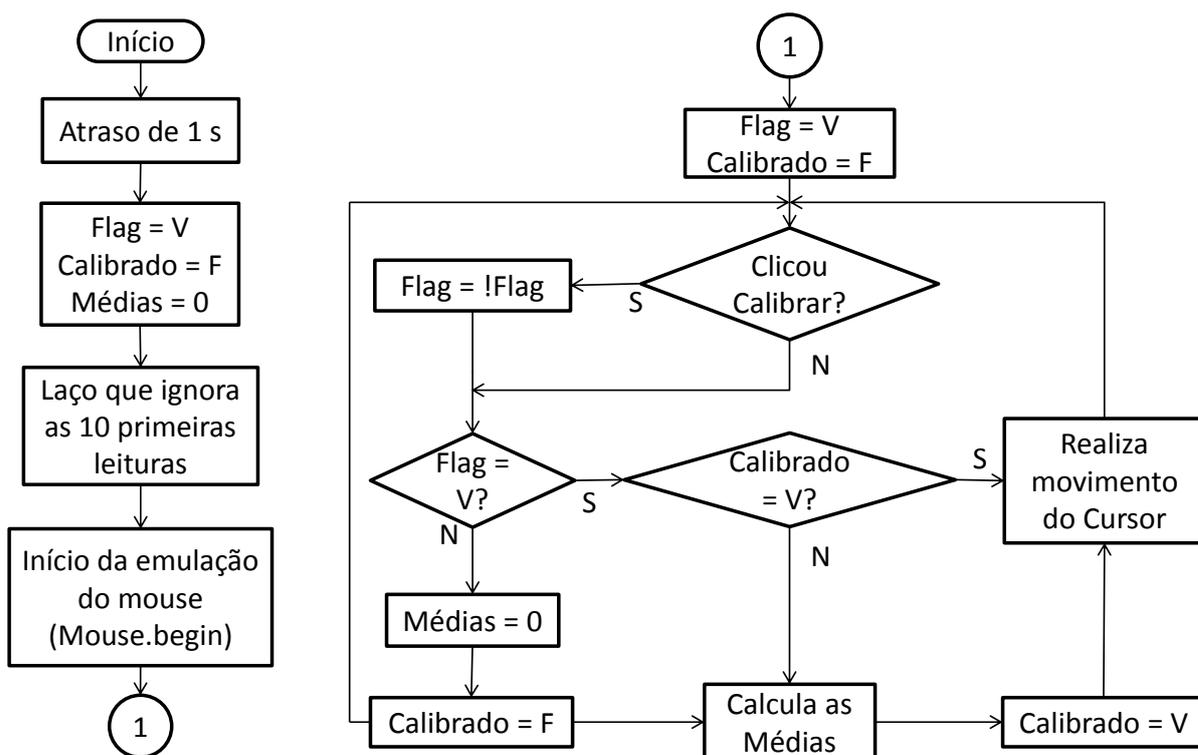


Figura 2.11 Fluxograma do código Arduino

Martin Millmore sugere que o dispositivo que ele desenvolveu pode ser usado em pessoas com deficiências motoras, sendo o dispositivo acoplado a qualquer parte do corpo, cabeça, pés ou mãos, por exemplo.

A necessidade de clique do mouse é um ponto importante a se considerar, quando o contexto é alguma deficiência motora. A solução depende do nível de mobilidade do usuário do dispositivo. No caso desse projeto, Martin desenvolveu um clique estacionário, se o cursor do mouse mover-se dentro de uma área de vinte pixels quadrados num tempo de um segundo, o clique acontece. Caso o cursor saia desta área determinada, o contador de tempo é reiniciado.

O custo estimado para a execução do projeto de Martin está na Tabela 2.1, em cotação no dia 24/08/2017, com preços obtidos nas lojas AliExpress [27] e Banggood [28].

Tabela 2.1 - Tabela de preços dos componentes eletrônicos utilizados

Componente	Quantidade	Preço (R\$)
ATmega32U4 Pro Micro 5 V 16 MHz	1	12,60
LSM9DS0	1	39,97
Conversor Lógico	1	1,50
Total		54,07

Fonte: AliExpress e Banggood

Capítulo 3

Materiais e Métodos

A realização deste trabalho iniciou-se pela pesquisa de referências sobre TA, usando a base Scopus [29] e outros repositórios disponíveis na Internet. Na sequência pesquisou-se as especificações e características dos componentes de hardware utilizados no projeto do *headmouse* de Millmore.

Posteriormente estudou-se a linguagem Arduino e as bibliotecas de código necessárias ao projeto. Assim, o código desenvolvido foi integrado ao protótipo de hardware construído no Laboratório de Sistemas Computacionais (LSC) do curso de Ciência da Computação da UNIOESTE Cascavel.

Com software e hardware integrados, iniciou-se a etapa de avaliação das funcionalidades e precisão na detecção dos movimentos do dispositivo. A cada falha ou defeito detectado, realizou-se reprojeção do hardware ou do software visando corrigi-los.

Realizadas as avaliações e ajustadas as funcionalidades, iniciaram-se os aperfeiçoamentos do projeto. Desenvolveu-se um sensor de mordida, que emula os cliques do mouse de acordo com o número de mordidas efetuadas, e um software cliente para alternar entre modo “leitura”, onde apenas o movimento do cursor permanece habilitado, ou modo “interação”, onde movimento e clique são habilitados; o clique pode ser configurado para clique simples, duplo ou com o botão direito.

Concluídas as melhorias, iniciou-se a fase de testes para validação das mesmas. Realizaram-se testes com a discente, no Centro de Reabilitação Física da UNIOESTE, e com o autor do projeto. Estes testes estão detalhados na seção 4.2 deste documento.

3.1 Componentes eletrônicos

3.1.1 Conversor lógico bidirecional

O conversor de nível lógico bidirecional possibilita a passagem de dados na direção HIGH para LOW e/ou LOW para HIGH. Como a interface I²C utiliza apenas um fio de transmissão de dados, o conversor bidirecional funciona perfeitamente.

Para entendermos mais sobre o funcionamento do conversor bidirecional, providos do esquemático do componente (Figura 3.1), faremos uma análise mais aprofundada. Ao final deste tópico, entenderemos como as entradas e saídas do componente funcionam.

O esquemático nos mostra um dos quatro canais idênticos, cada um contém um circuito bi-direcional de transmissão de tensão. Cada circuito contém um transistor BSS138, que serve como chave, e um par de resistores *pull-up* [30] para evitar flutuação, garantindo os níveis lógicos esperados.

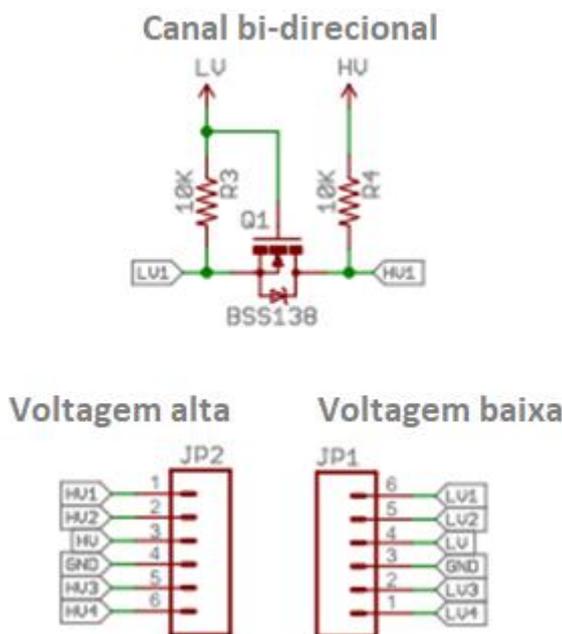


Figura 3.1: Esquemático logic level converter

Fonte: Modificado de Sparkfun [23]

São no total doze pinos no conversor – duas linhas paralelas com seis conexões, que podem ser visualizadas na Figura 3.2. A linha de alta tensão (linha superior) contém todas as entradas e saídas em 5 V; a linha de baixa tensão (linha inferior) contém todas as entradas em 3.3 V.

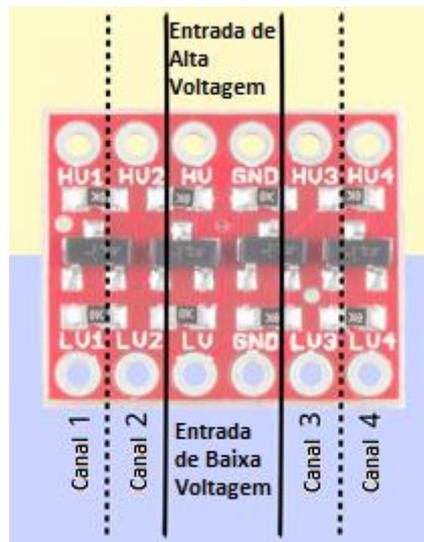


Figura 3.2: Divisão de canais do conversor

Fonte: Modificado de Sparkfun [31]

Dois pinos com rótulo HV e LV provêm as altas e baixas voltagens de referência para a placa. Os pinos GND são os pinos de referência para a tensão de zero volt. É necessário que sejam fornecidas tensões constantes e estáveis para todas as entradas citadas.

Os quatro canais de dados são separados no conversor, cada um deles é capaz de transmitir dados do lado alto para o lado baixo e vice-versa. Esses pinos são rotulados com HV1, LV1, HV2, LV2, HV3, LV3, HV4 e LV4. O número no rótulo corresponde ao canal do pino, e o prefixo HV corresponde à alta tensão, logo LV corresponde à baixa tensão.

3.1.2 Divisor de tensão

Como a tensão de referência para o nível baixo do conversor lógico é 3.3 V, e a única tensão que o microcontrolador fornece é de 5 V, torna-se necessário o uso do divisor de tensão.

A técnica de divisão de tensão consiste em estabelecer uma proporcionalidade entre as tensões V_{in} e V_{out} .

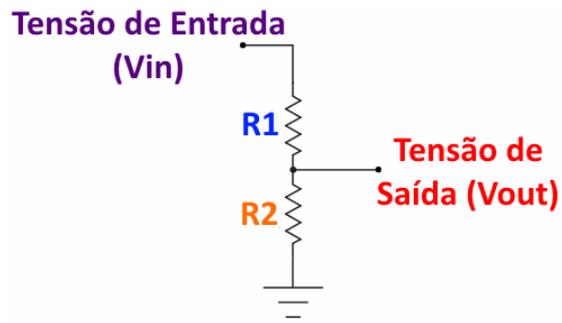


Figura 3.3: Divisor de tensão

Fonte: Arduino e Cia [32]

Dois resistores são conectados em série como na Figura 3.3 acima. Então temos que a tensão de saída (V_{out}) é dada pela equação 1²:

$$V_{out} = R_2 / (R_1 + R_2) * V_{in} \quad (\text{eq. 1})$$

Neste projeto (Figura 3.4) foi usado $R_1 = 220 \Omega$, $R_2 = 47 \Omega$, $V_{in} = 5 \text{ V}$ e $V_{out} = 3.3 \text{ V}$.

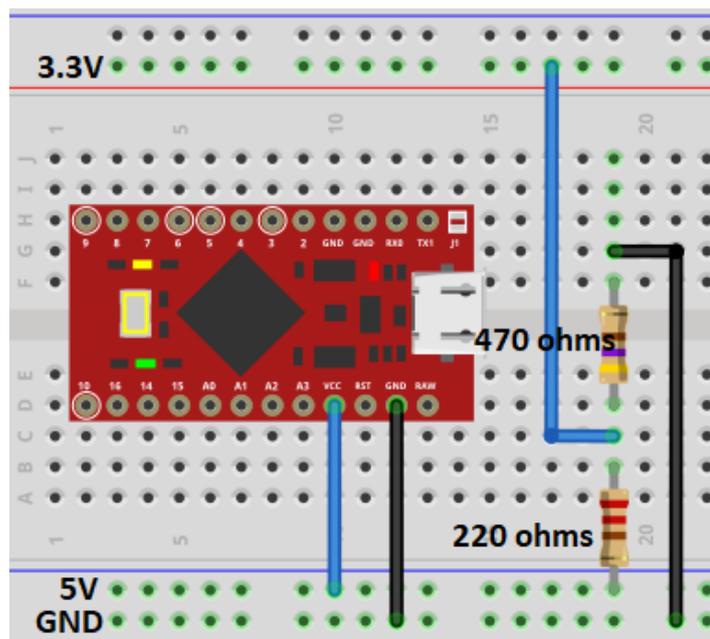


Figura 3.4: Esquema do conversor de tensão

² Equação deduzida a partir da Lei de Ohm [45]

3.1.3 LSM9DS0

O LSM9DS0 (Figura 3.6) é um componente de sensor de movimento que contém um acelerômetro, giroscópio e um magnetômetro, todos de três eixos (X, Y e Z). O componente tem suporte para as comunicações I²C e SPI.

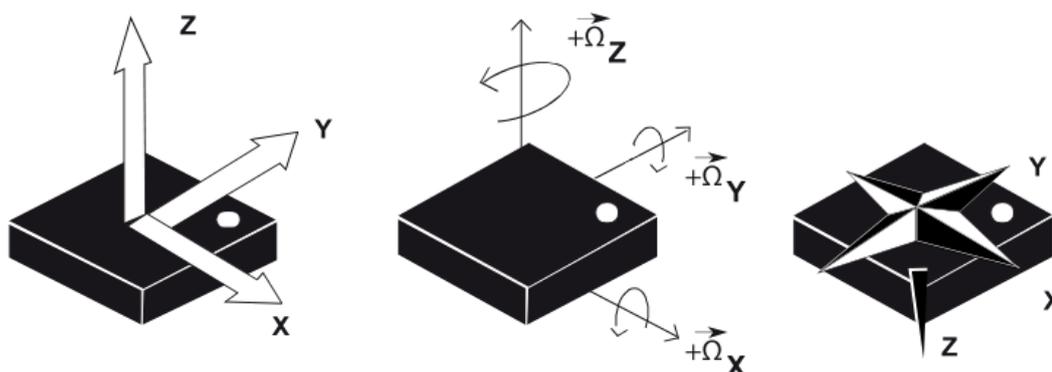


Figura 3.5 Representação visual dos três sensores (acelerômetro, giroscópio e magnetômetro)

Fonte: Sparkfun [22]

O giroscópio pode medir a velocidade angular (graus/s), que é a velocidade em que o componente está girando no espaço. O acelerômetro é capaz de medir a aceleração (m/s^2), a qual indica o quão rápido é a mudança de velocidade do componente. Finalmente, o magnetômetro, que pode mensurar a força e direção de campos magnéticos (Gs), e assim, se orientar no espaço. Neste trabalho, não se utilizou o magnetômetro.

A interface I²C foi escolhida para comunicação do componente por que o projeto original foi implementado primeiramente nessa interface. Outro ponto favorável é a redução do uso de fios para as conexões I²C.

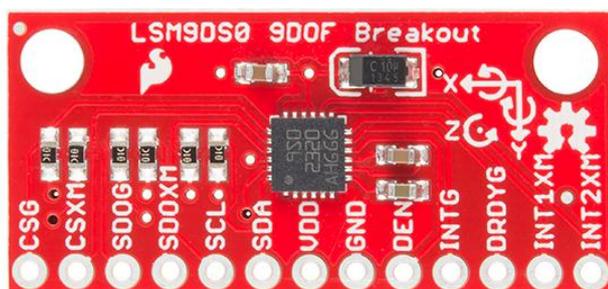


Figura 3.6: Sensor LSM9DS0

Fonte: Sparkfun [22]

No total, o LSM9DS0 possui 13 pinos. Serão descritos apenas os pinos que foram usados no projeto (SCL, SDA, VDD, GND, DRDYG INT1XM e INT2XM), para mais informações sobre os demais pinos, consulte o guia fornecido pela Sparkfun [22].

Os pinos SCL e SDA referem-se à comunicação I²C, o pino SCL é o clock serial e o pino SDA refere-se à transmissão bidirecional de dados serial. Estes pinos possuem um resistor *pull-up*. VDD e GND são as referências de tensão para o componente. O VDD é a tensão de alimentação, deve ser regulada entre 2,4 V e 3,6 V. O GND é a referência em 0 V.

O pino DRDYG gera uma interrupção que indica quando um novo dado do giroscópio está pronto, ou quando o *buffer* de dados está cheio. INT1XM e INT2XM são ambos uma interrupção programável, que pode enviar dados prontos, ou em sequência. O dorso do componente possui três *jumpers* de dois caminhos, como pode ser visualizado na Figura 3.7. Cada um desses jumpers é fechado com o propósito de, automaticamente, colocar o sensor LSM9DS0 em modo I²C.

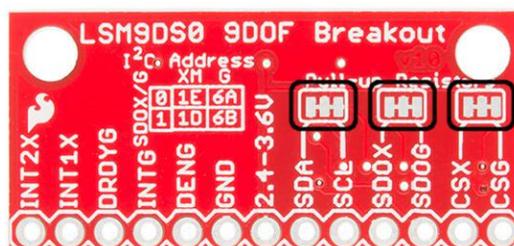


Figura 3.7: Três jumpers do LSM9DS0 no dorso do componente

Fonte: Sparkfun [22]

Cada um destes jumpers junta um par de pinos ao VDD, passando por um resistor de 10k Ω , como pode ser visualizado no esquemático abaixo:

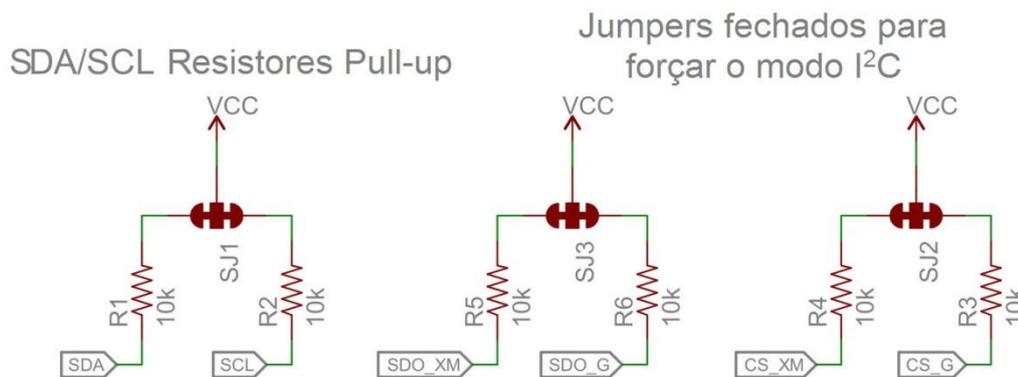


Figura 3.8: Esquemático dos Jumpers do LSM9DS0

Fonte: Sparkfun [21]

O *jumper* mais à direita conecta os pinos CSG ao pino CSX, através de um resistor de *pull-up*, estabelecendo a comunicação com o sensor no modo I²C. O *jumper* do meio conecta o pino SDOG ao SDOX, nos quais se situam os endereços do componente. Por último, o *jumper*, mais à esquerda, conecta os pinos SDA e SCL.

A intenção dos *jumpers* é facilitar o uso do componente, fazendo com que menos fios sejam necessários para fazê-lo funcionar. Para usar a comunicação I²C, pode-se ignorar os pinos com prefixo CS e SDO.

Para desabilitar qualquer um desses *jumpers*, deve-se fazer um corte nas pequenas conexões entre os *jumpers* e os pinos.

3.1.4 Pro Micro Atmega32U4

O Arduino Pro Micro ATmega32U4 5 V/16 MHz é um microcontrolador clone do Arduino Leonardo, compatível com Arduino. O transceptor USB, dentro do 32U4, permite uma conexão USB programável para agir como um mouse, teclado ou qualquer dispositivo USB.

Todas as entradas, saídas e alimentação do Pro Micro são divididas em duas linhas paralelas. Como pode ser visualizado na Figura 3.9.

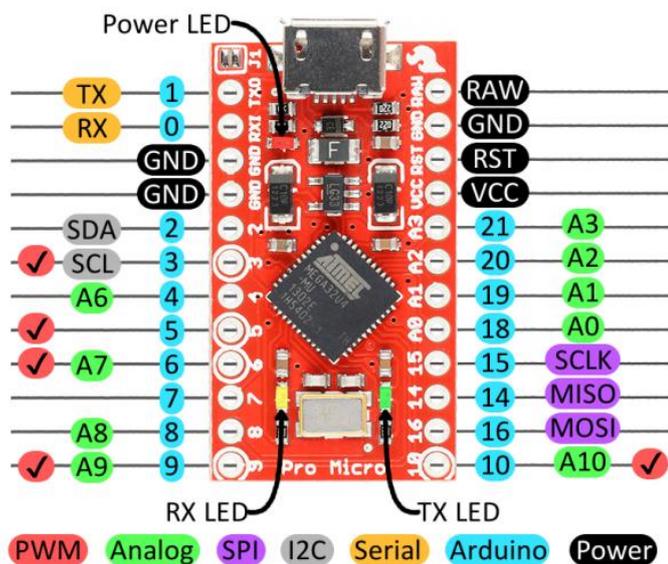


Figura 3.9: Pinagem do Arduino Pro Micro

Fonte: Sparkfun [33]

Neste trabalho serão descritos apenas os pinos que são usados no projeto (2, 3, 5, 6, 7, 10, VCC e GND). Para mais informações e detalhamento sobre os demais pinos do componente, vide Sparkfun Pro Micro *Hookup Guide* [33] e Pro Micro *schematic* [34].

VCC é o pino de alimentação da placa. A tensão depende do tipo de Pro micro usado, 3.3 V/8 MHz ou 5 V/16 MHz. No caso deste projeto, utiliza-se a 5 V/16 MHz. Como a alimentação da placa é feita por USB, o VCC fornece aproximadamente 4.68 V. o GND é a referência de 0 V.

Os pinos numerados de 0 a 21 (incluindo os pinos 5, 6, 7 e 10) possuem diversas funções e podem ser de entrada ou saída digital. No caso desse projeto, os pinos empregados para comunicação (2 e 3) utilizam o protocolo I²C.

Capítulo 4

Resultados e Discussão

Nesta seção são apresentadas as soluções proposta, as interpretações e apontamentos a partir dos resultados obtidos durante o desenvolvimento do projeto.

O protótipo do *headmouse* foi desenvolvido conforme o especificado no projeto de Martin Millmore [5] e, durante a avaliação das funcionalidades e precisão do projeto, identificou-se uma dificuldade relacionada ao clique do mouse. Este acontece quando o usuário permanece, por um segundo, com o cursor do mouse relativamente estático. Assim, durante a leitura de algum texto ou visualização de vídeo em tela cheia, vários cliques do mouse são efetuados, pois há pouco movimento do cursor.

Algumas modificações e aprimoramentos do projeto original foram feitos: o divisor de tensão, o sensor de mordida e o software cliente, que serão descritos na sequência.

4.1 O protótipo do Headmouse

A Figura 4.1 apresenta o primeiro protótipo do Headmouse desenvolvido, contendo suas modificações de hardware, sendo estas descritas nos subcapítulos subsequentes do texto.

O protótipo final (Figura 4.2), concedido à discente citada anteriormente, não contém o sensor de mordida, dado que a discente não tem controle dos movimentos da boca. O circuito encontra-se protegido por uma caixa moldada a partir de um material termoplástico, sendo fixado à cabeça por meio de uma faixa de elástico com uma fivela, como apresentado na Figura 4.3.

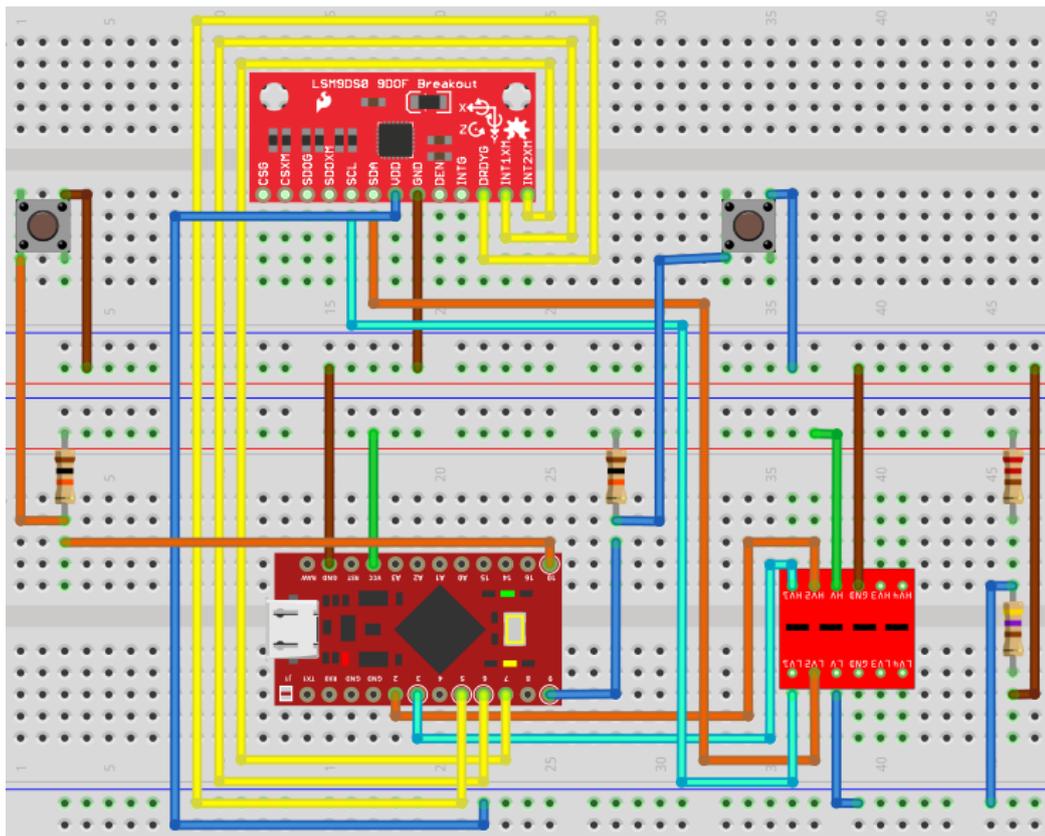


Figura 4.1: Projeto do primeiro protótipo do Headmouse



Figura 4.2: Protótipo final do Headmouse.



Figura 4.3: Fixação do protótipo na cabeça

4.1.1 Divisor de tensão

A modificação do divisor de tensão foi necessária, pois Martin, em seu projeto original, usou um microcontrolador que tinha fornecimento de tensão em 3.3 V, enquanto o microcontrolador usado neste trabalho fornece apenas tensão em 5 V.

4.1.2 Sensor de mordida

Para solucionar a dificuldade relacionada ao clique do mouse, desenvolveu-se um sensor de mordida, como mostra a Figura 4.4. Ao morder um botão (Figura 4.5), conectado à porta digital 9 do Arduino, o usuário efetua o clique do mouse.



Figura 4.4: Sensor de mordida

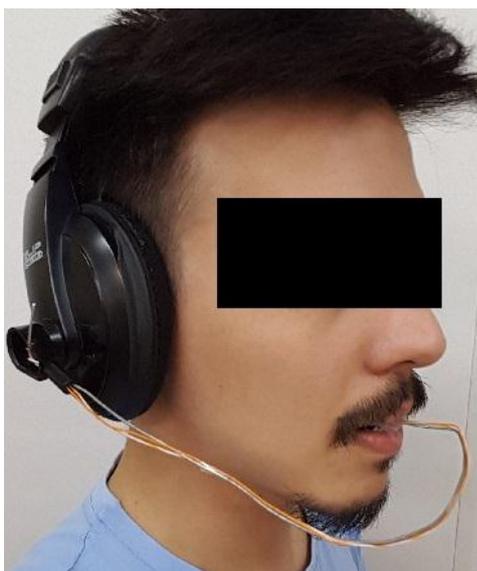


Figura 4.5: Uso do sensor de mordida

A lógica do código do sensor está representada na máquina de estados apresentada na Figura 4.6.

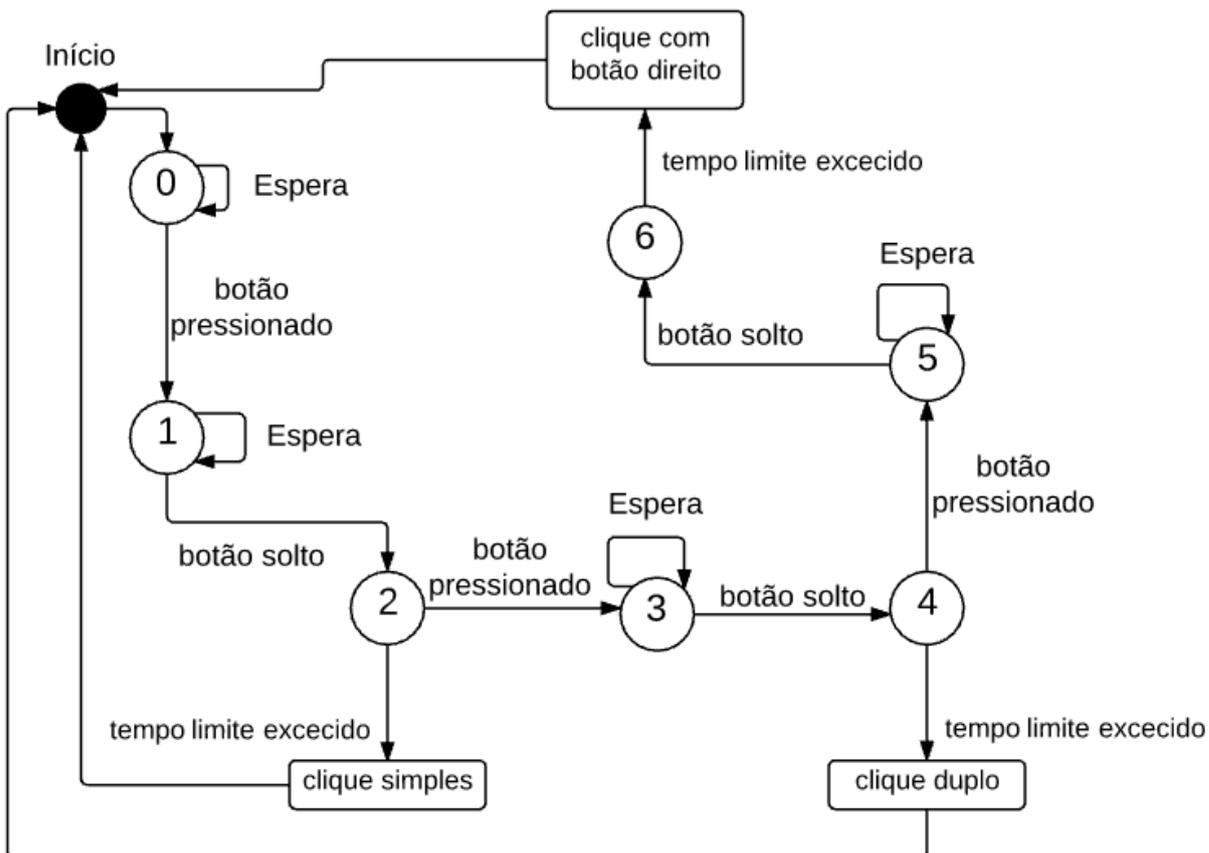


Figura 4.6: Diagrama de Máquina de Estados do Clique do Mouse

A máquina inicia-se no estado 0; quando o botão é pressionado passa-se para o estado 1, que consiste na espera até que o botão seja solto, gerando a transição que leva ao estado 2. No estado 2 verifica-se o tempo decorrido desde o estado 0; se o tempo limite for atingido, o clique simples é efetuado e a máquina retorna ao estado inicial. Se o tempo não é excedido e o botão é pressionado novamente, temos a transição para o estado 3. Agora no estado 3 a lógica se repete, ficando em espera até que o botão seja solto; quando isso acontecer passamos para o estado 4. O estado 4 segue a mesma lógica do estado 2; se o tempo limite for excedido, dessa vez o clique duplo é efetuado e a máquina retorna ao estado inicial; caso contrário, um novo clique no botão leva ao estado 5, que também é um estado de espera; quando o botão é solto novamente, passa para o estado 6. Diferente dos estados 2 e 4, a única transição de estado corresponde ao tempo limite que, ultrapassado, efetua o clique com o botão direito e a máquina retorna ao estado inicial. A implementação da máquina de estados que implementa o clique do mouse encontra-se em detalhes no anexo B deste trabalho.

Com os testes realizados com o autor deste projeto, o sensor de mordida mostrou-se adequado, realizando todos os tipos de clique com facilidade.

4.1.3 Software cliente

Como a discente que inspirou este Trabalho de Conclusão de Curso não tem os movimentos da boca, o sensor de mordida não a auxilia na interação com o computador. Para contornar esta limitação fisiológica, desenvolveu-se um software cliente em Java [35] que, através de comandos enviados pela porta serial (USB), comunica-se com a placa Arduino, alterando assim o tipo de clique que será efetuado pela usuária. O projeto de Martin possui rotinas para clique automático, que é executado quando o cursor fica estacionado por 1 segundo em uma área de 20x20 pixels quadrados; o software cliente desenvolvido permite ativar/desativar este clique automático e personalizá-lo como clique simples, duplo ou como botão direito do mouse.

O software cliente e a placa Arduino trocam informações usando comunicação serial. Todas as placas Arduino tem pelo menos uma porta serial [36]. A comunicação serial pode ser realizada através dos pinos digitais 0 (RX) ou 1 (TX) ou através do conector USB.

A biblioteca RXTX [37] permite a uma aplicação Java comunicar-se através da porta serial com dispositivos periféricos. Basicamente a aplicação Java, usando a RXTX, cria uma conexão com a porta serial pré-configurada. Quando a conexão é estabelecida com sucesso,

informações são enviadas com uma função de escrita na porta serial. Esses comandos podem ser visualizados no método initialize() apresentado na Figura 4.7.

```
private void initialize() {
    try {
        //Define uma variável portId do tipo CommPortIdentifier para realizar a comunicação serial
        CommPortIdentifier portId = null;
        try {
            //Tenta verificar se a porta COM informada existe
            portId = CommPortIdentifier.getPortIdentifier(this.portaCOM);
        } catch (NoSuchPortException npe) {
            //Caso a porta COM não exista será exibido um erro
            JOptionPane.showMessageDialog(null, "Porta COM não encontrada.",
                "Porta COM", JOptionPane.PLAIN_MESSAGE);
        }
        //Abre a porta COM
        SerialPort port = (SerialPort) portId.open("Comunicação serial", this.taxa);
        serialOut = port.getOutputStream();
        port.setSerialPortParams(this.taxa, //taxa de transferência da porta serial
            SerialPort.DATABITS_8, //taxa de 10 bits 8 (envio)
            SerialPort.STOPBITS_1, //taxa de 10 bits 1 (recebimento)
            SerialPort.PARITY_NONE); //receber e enviar dados
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 4.7: Método initialize

A aplicação Java contém cinco botões e uma caixa de combinação (Figura 4.8). No lado esquerdo desta figura vemos o botão indicando o modo Leitura; quando este botão é pressionado alterna-se para o modo Interação (Clique) conforme observado no lado direito desta figura. O botão “Simples” define que o clique automático passará a ser o clique simples com o botão esquerdo do mouse, o botão “Duplo” define que o clique automático será o clique-duplo com o botão esquerdo do mouse e o botão “Direito” define que o clique automático será o clique com o botão direito do mouse. Por fim, o botão “Calibrar” recalcula as médias de calibração do headmouse.

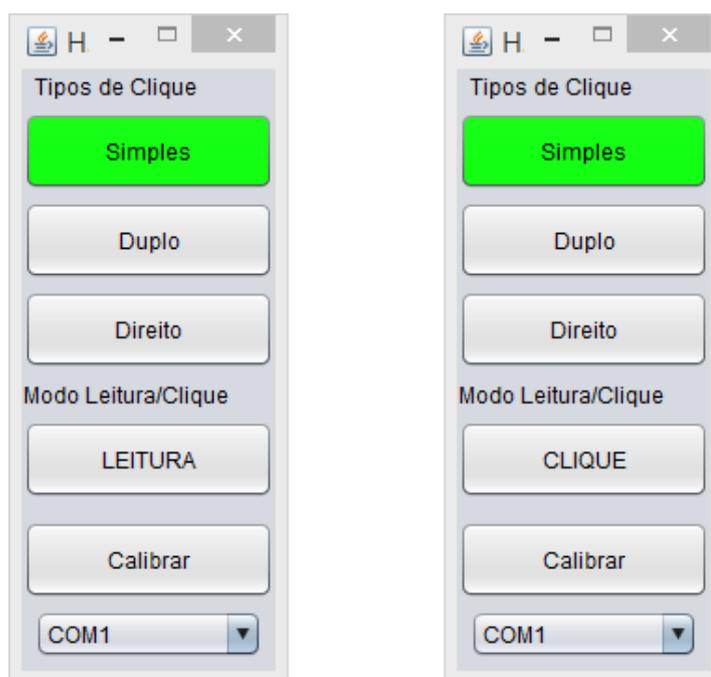


Figura 4.8: Aplicação Java para comunicação Serial

Já a caixa de combinação seleciona em qual porta COM o dispositivo headmouse está conectado. Deve-se selecionar a porta correta para que haja comunicação do dispositivo com a aplicação Java.

Todos os botões, exceto “Calibrar” e a caixa de combinação, são ativados pelo evento *Mouse_Entered*, que é disparado quando o cursor passa sobre a área do botão. O botão “Calibrar” responde ao evento *Mouse_Clicked*, pois assim evita-se que o mouse seja calibrado acidentalmente pelo usuário. A caixa de combinação responde ao evento *Popup_Menu_Will_Become_Invisible*, quando a lista de itens da caixa se contrai, o item selecionado é ativado. Para ser calibrado corretamente o *headmouse* precisa estar estático sobre uma superfície plana.

4.2 Testes Realizados

A versão do *headmouse*, sem a opção para diferentes tipos de cliques, foi testada com a paciente citada neste projeto, juntamente com o teclado virtual MouseKey. O MouseKey implementa, nativamente, o clique automático quando o cursor permanece sobre um botão por um certo tempo, que é configurado na aplicação.

A paciente apresentou dificuldade em controlar o cursor, pois além da tetraplegia ela tem sua coordenação motora fina comprometida; ou seja, apresenta dificuldade para manter o

cursor do mouse parado na tela. Para superar esta limitação diminuiu-se a sensibilidade do cursor do mouse no painel de controle do Sistema Operacional Windows; esta medida simples auxiliou a paciente diminuindo o tremor do cursor, porém acarretou aumento na amplitude dos movimentos da cabeça para movimentar o mouse. A solução adotada para reduzir a amplitude dos movimentos de cabeça foi diminuir a resolução da tela do computador, o que contribuiu para reduzir os problemas.

Mesmo com estas medidas complementares, ainda persistiu um pequeno tremor do cursor por conta dos movimentos involuntários da paciente. À medida que o tempo de uso foi aumentando, melhoras no controle do cursor foram notadas, a ponto de a paciente conseguir escrever seu nome completo e algumas frases.

Em relação ao uso do software MouseKey, duas características deste software afetam a usuária. A primeira característica diz respeito ao tamanho do botão de sílabas que ficam no entorno das teclas principais do teclado virtual (Figura 4.9); como elas são pequenas, a usuária não consegue manter o cursor sobre a área destas teclas pelo tempo necessário para que o clique automático seja efetuado. A segunda característica diz respeito à falta de retorno visual vinculado ao clique automático; a usuária não consegue perceber se o clique automático foi efetuado, principalmente quando a intenção é adicionar um espaço em branco, pois não há ponto de inserção que indica onde será inserido o caractere.



Figura 4.9: Tela da letra “b” do teclado virtual MouseKey

Fonte: NIEE UFRGS [4]

Testes com o sensor de mordida foram feitos apenas com o autor deste projeto. O sensor apresentou-se muito eficaz para efetuar os cliques de todos os tipos. Pode-se perceber que posicionar o sensor na boca sem o uso das mãos torna-se uma tarefa difícil, necessitando de auxílio de terceiro.

Os testes do software cliente também foram feitos apenas com o autor deste projeto. O Software oferece todos os recursos prometidos e seu uso é intuitivo. Alguns problemas puderam ser observados: a janela da aplicação foi fixada no canto superior esquerdo, sobre quaisquer outras janelas de aplicação que estejam abertas no computador, reduzindo sua

visibilidade; outro problema é a ativação não intencional das funcionalidades do software, por exemplo, ativação do clique duplo, pois algumas das funcionalidades são ativadas pelo evento *Mouse_Entered*.

Capítulo 5

Considerações Finais

Pessoas com mobilidade dos braços reduzida têm dificuldade em comandar o cursor do mouse. Neste trabalho desenvolveu-se um *headmouse* que, utilizando movimentos da cabeça, permite ao usuário controle sobre o cursor do mouse. Baseado no projeto de Martin Millmore, desenvolveu-se um software cliente que possibilita ao usuário controlar os movimentos do cursor, quando no modo “leitura”, e também efetuar cliques simples, duplos e com o botão direito, quando no modo “clique”. Para usuários tetraplégicos que controlam sua mordida, acrescentou-se um sensor que emula os cliques do mouse de acordo com o número de mordidas efetuadas.

O *headmouse* desenvolvido foi testado com a paciente alvo deste projeto e, com base neste teste, pôde-se observar que os objetivos deste trabalho foram alcançados. O dispositivo funciona como um mouse, permitindo controlar os movimentos do cursor e realizar as ações (cliques) necessárias, seja através do sensor de mordida desenvolvido ou através do software cliente que permite alternar entre modo leitura ou interação, alternando entre os diferentes tipos de clique. Ressalta-se que a TA desenvolvida abre um leque de opções para seu aperfeiçoamento, como uso de filtros para reduzir o efeito do tremor dos pacientes ou outros tipos de sensores para efetuar os cliques do mouse.

Comparando o *headmouse* desenvolvido com outros dispositivos atualmente comercializados, esse se apresenta viável em termos de custo e implementação. Os componentes que integram o projeto têm baixo custo e sua construção é simples, seguindo-se os passos descritos neste trabalho de conclusão de curso.

Na realização do presente trabalho observou-se a importância e o impacto dos benefícios que o dispositivo *headmouse* pode trazer a um portador de deficiência física, que tenha comprometidas as funções motoras em seus membros superiores. Assegura-se, com toda certeza, que esta TA pode conferir maior autonomia, acesso aos meios de comunicação e a fontes de informação digital, contribuindo inclusive com a ampliação da interação social das

pessoas com deficiência com os demais membros da sociedade.

Vale ressaltar que, apesar de simples, o sistema implementado favorece a independência dos tetraplégicos no uso de computadores compatíveis com o *headmouse*. Pequenas mudanças em situações rotineiras, como poder se comunicar sem o auxílio de terceiros, contribuem para suavizar o impacto de suas limitações físicas, deixando-os mais confortáveis e adaptados à sua própria realidade. O *headmouse* desenvolvido abre portas, torna o estigma da tetraplegia menos impactante na vida das pessoas com deficiência e acrescenta a sensação de bem-estar àqueles envolvidos em seu projeto.

5.1 Trabalhos Futuros

Como foi possível observar nos testes, a usuária do protótipo apresentou dificuldades em manter estáveis seus movimentos de cabeça; uma possível solução computacional a ser avaliada em trabalho futuro é a aplicação de um algoritmo de suavização, como por exemplo o filtro complementar [38] ou o filtro de Kalman [39]. O primeiro é um filtro passa baixa que consiste em uma combinação das leituras do acelerômetro e do giroscópio para suavizar as leituras. O segundo permite estimar dados de um sistema dinâmico suavizando os seus ruídos.

Considerando as dificuldades encontradas nos testes em relação ao software MouseKey, elaborou-se algumas sugestões para modificações do mesmo, sendo elas: ao focar uma determinada letra ampliá-la juntamente com as sílabas no seu entorno. Quanto ao clique automático do MouseKey, um retorno visual, que consiste na alteração momentânea da cor de fundo do botão, ou auditivo (bipe), evidenciaria o clique efetuado.

Tendo em vista os testes realizados, recomenda-se testar o *headmouse* com mais usuários com deficiência, principalmente testes relacionados ao software cliente e ao sensor de mordida.

Quando ao projeto de hardware algumas sugestões são: a substituição do sensor LSM9DS0 pelo sensor mpu6050 [40], que têm saídas e entradas digitais que funcionam em 5 V, ou seja, tornando desnecessário o uso do conversor lógico e do divisor de tensão e, conseqüentemente, diminuindo o tamanho do projeto de hardware. Outro ponto a se considerar é o barateamento do projeto, já que o sensor mpu6050 é muito mais barato que o LSM9DS0. O custo do primeiro é de aproximadamente R\$ 3,39 [41], em cotação no dia 28/11/2017, contra R\$ 39,97 [42] do segundo, em cotação no dia 24/08/2017.

Anexo A

A.1 Preparação do Ambiente Windows/Arduino Pro Micro

Para que o Arduino Pro Micro 32U4 seja reconhecido no Windows e no ambiente Arduino é necessária instalação de *drivers*. Segue abaixo um passo-a-passo de como é feita a instalação e reconhecimento do componente em questão.

A.1.1 Instalação do *Driver Sparkfun Pro Micro*

O primeiro passo é fazer *download* do *driver* diretamente do GitHub Repository [43]. Efetuado o *download*, é preciso descompactar o arquivo. O *driver* encontra-se em `Arduino_Boards-master/sparkfun/avr/signed_driver`. Ao conectar o Arduino Pro Micro ao computador, via USB, surgirá a mensagem do sistema operacional “*Installing device driver software*”, como na Figura .

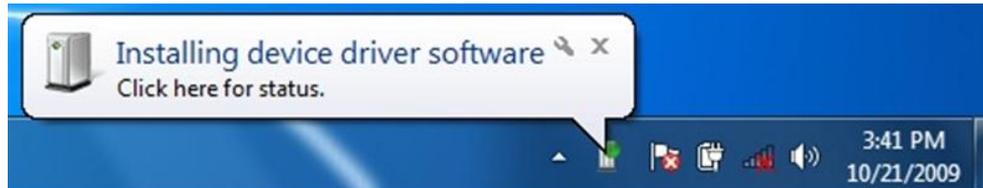


Figura A.1 “*Installing device driver software*”

Na sequência surgirá outra mensagem: “*Device driver software was not successfully installed*”. Isso significa que o Windows não conseguiu encontrar um driver para o dispositivo.

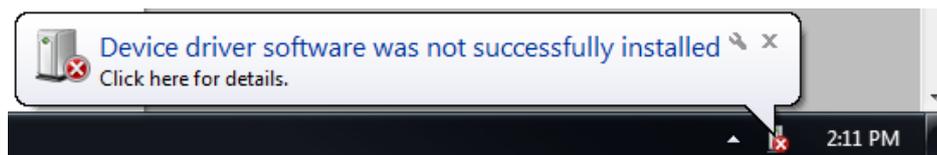


Figura A.2 “*Device driver software was not successfully installed*”

Agora será necessário abrir o *Device Manager*; tecle “*Windows key + R*” e digite “*devmgmt.msc*” e em seguida clique em “*OK*”, conforme ilustrado em Figura .

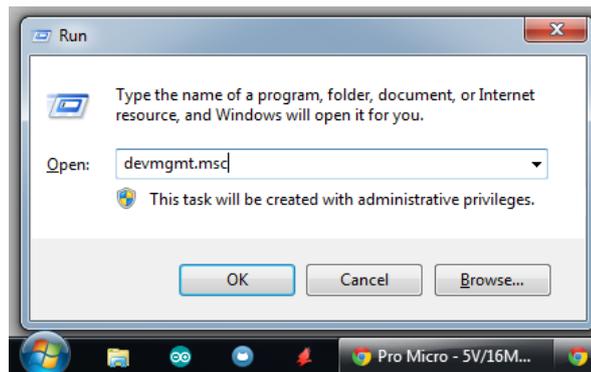


Figura A.3 Prompt de comando

No *Device Manager* expanda a árvore “*Other Devices*”, localize o dispositivo “*USB IO Board*” com indicador de alerta (triângulo amarelo com sinal de exclamação). Clique com o botão direito do mouse sobre o dispositivo e selecione “*Update Driver Software*”.

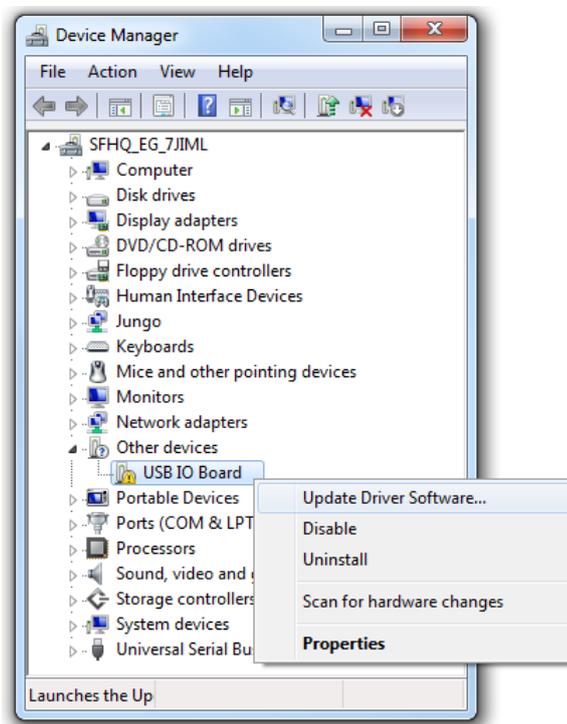


Figura A.4 *Update Driver Software*

Na primeira janela que surgir clique “*Browse my computer for driver software*”; na próxima janela, clique em “*Browse...*” e encontre o *driver* baixado. Depois de selecionada a pasta do *driver*, clique em “*OK*” e depois “*Next*”.

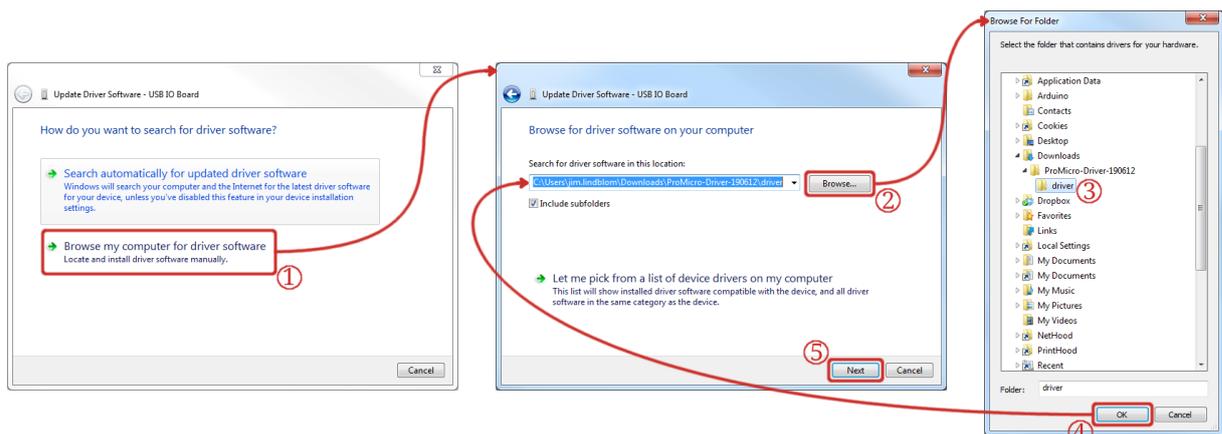


Figura A.5 “Update Driver Software”

O Windows avisará que o *driver* não é assinado digitalmente. Prossiga a instalação clicando em “*Install this driver anyway*”. Quando a instalação for concluída no *Driver Manager* será possível visualizar o dispositivo “*Sparkfun Pro Micro (COM ##)*”.

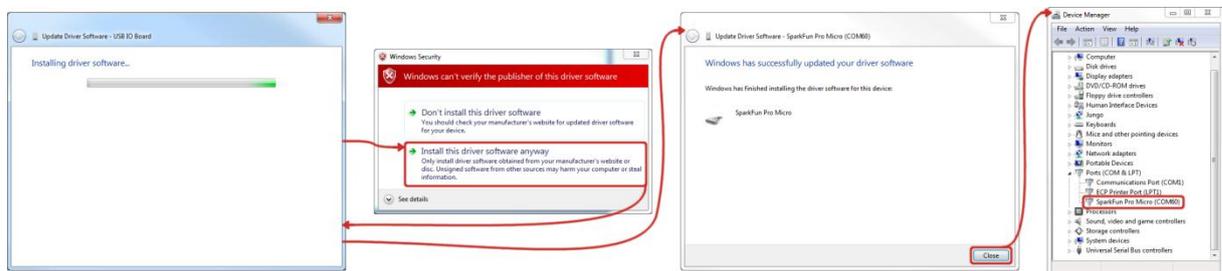


Figura A.6 Nova entrada do *DriverManager*

A.1.2 Adicionando o Sparkfun Pro Micro à Arduino IDE

Este é o último passo antes de poder usar a Arduino Pro Micro 32U4. Antes deste passo é necessário garantir que a Arduino IDE esteja instalada, link para download pode ser encontrado no site oficial Arduino [44]. É necessário instalar os arquivos (.brd), que possibilitam a comunicação entre a Arduino IDE e o microcontrolador.

Para começar indique à *Arduino IDE Board Manager* a URL (https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json) para que o *download* dos arquivos seja possível. Abra a IDE e clique em *File/Preferences*. Na parte de baixo da janela há um campo “*Additional Boards Manager URLs*”, cole a URL citada neste campo e clique em “*OK*”.

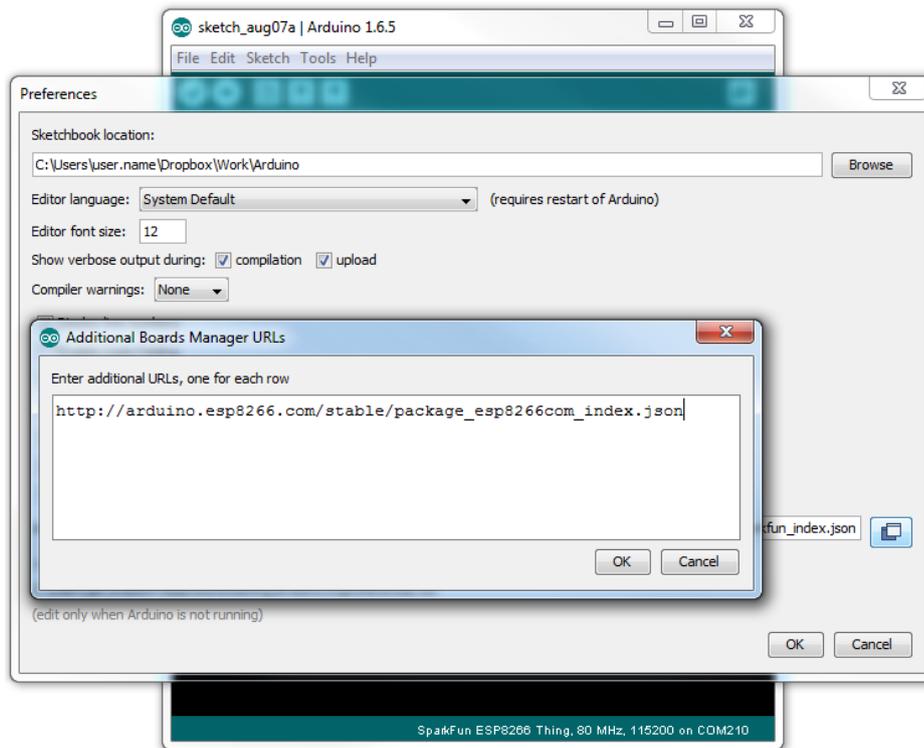


Figura A.7 *Additional Boards Manager URLs*

Na sequência clique em “*Tools/Board/Boards Manager...*”.

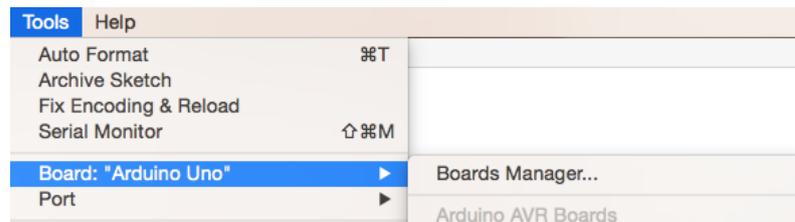


Figura A.8 *Tools/Board/Boards Manager...*

Faça uma busca por “*sparkfun*”. Localize o pacote da SparkFun (*SparkFun AVR Boards by SparkFun Electronics*), clique em *install*.

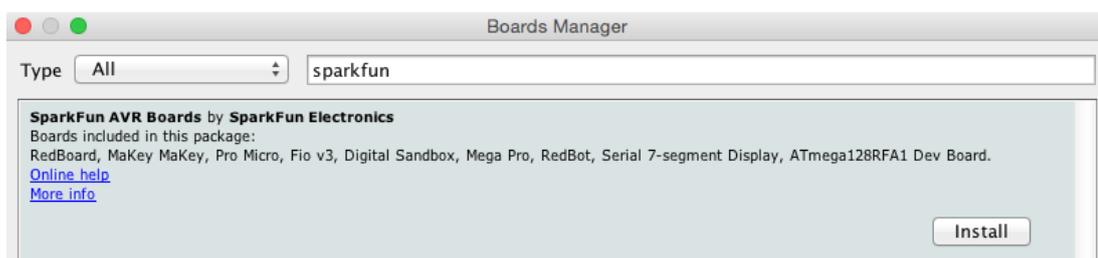


Figura A.9 *SparkFun AVR Boards by SparkFun Electronics*

Finalmente é possível carregar códigos a diversos tipos compatíveis aos produtos Arduino SparkFun, incluindo o Pro Micro.



Figura A.10 Produtos Arduinos SparkFun

Anexo B

B.1 Código da máquina de estados que implementa o clique do mouse com sensor de mordida

O código correspondente a transição do estado 0 para o estado 1 é apresentado na Figura B.1. Quando o botão é pressionado o estado é modificado para 1, uma variável de contagem de cliques é incrementada e a variável *clickTime* guarda o tempo atual; um atraso de 250 milissegundos é efetuado para garantir a leitura de apenas uma pressão do botão.

```
152 | switch(state){
153 |     case 0:
154 |         if(digitalRead(mouseClickButton) == LOW){
155 |             state = 1;
156 |             numberOfClicks++;
157 |             clickTime = millis();
158 |             delay(250);
159 |         }
160 |         break;
```

Figura B.1: Implementação do estado 0

Os estados 1, 3 e 5 são estados de espera e a transição para os respectivos estados 2, 4 e 6 ocorrem quando do botão é solto. O código corresponde a estas transições é apresentado na Figura B.2.

```
162 |     case 1:
163 |         if(digitalRead(mouseClickButton) == HIGH){
164 |             state = 2;
165 |         }
166 |         break;
167 |     case 3:
168 |         if(digitalRead(mouseClickButton) == HIGH){
169 |             state = 4;
170 |         }
171 |         break;
172 |     case 5:
173 |         if(digitalRead(mouseClickButton) == HIGH){
174 |             state = 6;
175 |         }
176 |         break;
```

Figura B.2: Implementação dos estados 1, 3 e 5

Os estados 2 e 4 seguem exatamente a mesma lógica; um condicional testa se o tempo decorrido desde o estado 0 é maior que o tempo pré-determinado. Se o tempo decorrido for maior, o estado é alterado para o estado 100 (Figura B.5). Se o tempo for menor e houver mais uma pressão no botão, o contador de cliques é incrementado, o estado é modificado para o próximo (3 se o estado for 2 e 5 se o estado for 4) e um atraso de 250 milissegundos é efetuado para garantir a leitura de apenas uma pressão do botão. A Figura B.3 mostra o código correspondente às transições dos estados 2 e 4.

```
168 |     case 2:
169 |         if(millis() - clickTime > dwellTime){
170 |             state = 100;
171 |         }else if(digitalRead(mouseClickButton) == LOW){
172 |             numberOfClicks++;
173 |             state = 3;
174 |             delay(250);
175 |         }
176 |         break;
182 |     case 4:
183 |         if(millis() - clickTime > dwellTime){
184 |             state = 100;
185 |         }else if(digitalRead(mouseClickButton) == LOW){
186 |             numberOfClicks++;
187 |             state = 5;
188 |             delay(250);
189 |         }
190 |         break;
```

Figura B.3: Implementação dos estados 2 e 4

O estado 6 possui apenas uma condição que testa se o tempo decorrido desde o estado 0 é maior do que o tempo pré-determinado. Se for maior o estado é alterado para o estado 100; senão permanece-se nesse estado até que seja atingido o tempo limite. A Figura B.4 mostra o código correspondente à transição do estado 6.

```
198 |     case 6:
199 |         if(millis() - clickTime > dwellTime){
200 |             state = 100;
201 |         }
202 |         break;
```

Figura B.4: Implementação do estado 6

Por fim, o estado 100 efetiva algum dos cliques do mouse; se o contador de números de clicks for 1, o clique simples é efetuado; se o valor do contador for 2, o clique duplo é efetuado; se o valor for 3, um clique com o botão direito acontece. Ao final reinicia-a o contador de cliques e a máquina retorna ao estado 0. A Figura B.5 mostra as ações e transições realizadas no estado 100.

```
204     case 100:
205         if(numberOfClicks == 1){
206             Mouse.click();
207         }
208         if(numberOfClicks == 2){
209             Mouse.click();
210             Mouse.click();
211         }
212         if(numberOfClicks == 3){
213             Mouse.click(MOUSE_RIGHT);
214         }
215         numberOfClicks = 0;
216         state = 0;
217         break;
218     }
```

Figura B.5: Implementação do estado 100

Referências

- [1] IBGE. **Censo Demográfico 2010 - Características Gerais da População - Resultados da Amostra.** Disponível em: <ftp://ftp.ibge.gov.br/Censos/Censo_Demografico_2010/Caracteristicas_Gerais_Religiao_Deficiencia/ods/Brasil_ods.zip>. Acesso em: 15 ago. 2017.
- [2] DUSIK, C. L.; COSTI SANTAROSA, L. M. **Mousekey.** Disponível em: <<http://www.ufrgs.br/niece/mousekey.php>>. Acesso em: 15 ago. 2017.
- [3] UNIVERSITAT DE LLEIDA. **Headmouse.** Disponível em: <<http://robotica.udl.cat/headmouse.htm>>. Acesso em: 15 ago. 2017.
- [4] DUSIK, C. L.; COSTI SANTAROSA, L. M. **Guia do Usuário.** Acesso em: 16 ago. 2017.
- [5] MILLMORE, M. **Head Mouse - Game Controller or Disability Aid.** Disponível em: <<http://www.instructables.com/id/Head-Mouse-Game-controller-or-disability-aid/>>. Acesso em: 15 ago. 2017.
- [6] ASSISTIVE TECHNOLOGY INDUSTRY ASSOCIATION. **What is AT?** Disponível em: <<https://www.atia.org/at-resources/what-is-at/>>. Acesso em: 17 ago. 2017.
- [7] STEVEN, L. **History of Assistive Technology Time Line.** Disponível em: <<https://www.timetoast.com/timelines/history-of-assistive-technology>>. Acesso em: 17 ago. 2017.
- [8] IBGE. **Vamos conhecer o Brasil|Nosso povo|Características da população.** Disponível em: <<http://7a12.ibge.gov.br/vamos-conhecer-o-brasil/nosso-povo/caracteristicas-da-populacao.html>>. Acesso em: 18 ago. 2017.
- [9] ROCHA RODRIGUES, P.; GAMA ALVES, L. R. TECNOLOGIA ASSISTIVA – UMA REVISÃO DO TEMA. **HOLOS**, 6, n. 29, 2013.
- [10] MEC/SECADI. **Política Nacional de Educação Especial na Perspectiva da Educação Inclusiva.** Disponível em: <http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=16690-politica-nacional-de-educacao-especial-na-perspectiva-da-educacao-inclusiva-05122014&Itemid=30192>. Acesso em: 18 ago. 2017.

- [11] HEINZEN AMIN HELOU, A. R. et al. **POLÍTICAS PÚBLICAS DE INCLUSÃO DIGITAL. Renote - Novas Eecnologias na Educação**, Porto Alegre, v. 9, n. 1, jul. 2011. ISSN 1679-1916.
- [12] PRESIDÊNCIA DA REPÚBLICA - CASA CIVIL. **D5296**, 2004. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/decreto/d5296.htm>. Acesso em: 18 ago. 2017.
- [13] MINISTÉRIO DOS DIREITOS HUMANOS. **Viver sem Limite - Ministério dos Direitos Humanos**. Disponível em: <<http://www.sdh.gov.br/assuntos/pessoa-com-deficiencia/programas/viver-sem-limite>>. Acesso em: 18 ago. 2017.
- [14] PEE. **PEE - Programa de Ações Relativas às Pessoas com Necessidades Especiais**. Disponível em: <<http://www5.unioeste.br/portal/proex/programas/pee/sobre>>. Acesso em: 31 out. 2017.
- [15] GLASSOUSE. **Glassouse Assistive Device**. Acesso em: 16 ago. 2017.
- [16] ORIGIN INSTRUMENTS. **HeadMouse Nano**. Acesso em: 16 ago. 2017.
- [17] BROADENED HORIZONS. **Broadened Horizons Direct - QuadMouse Consigurator**. Acesso em: 16 ago. 2017.
- [18] ORIGIN INSTRUMENTS. **Shop at Origin Instruments**. Acesso em: 16 ago. 2017.
- [19] ARDUINO. **Arduino - Introduction**. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 15 ago. 2017.
- [20] POLULU. **Polulu A-Star 32U4 micro**. Disponível em: <<https://www.pololu.com/product/3101>>. Acesso em: 17 ago. 2017.
- [21] SPARKFUN. **lsm9ds0-breakout**. Disponível em: <https://cdn.sparkfun.com/assets/8/c/c/4/9/lsm9ds0_breakout-v10-schematic-.pdf>. Acesso em: 17 ago. 2017.
- [22] SPARKFUN. **LSM9DS0 hookup guide**. Disponível em: <<https://learn.sparkfun.com/tutorials/lsm9ds0-hookup-guide>>. Acesso em: 16 ago. 2017.
- [23] SPARKFUN. **Logic_Level_Bidirectional**. Disponível em: <https://cdn.sparkfun.com/datasheets/BreakoutBoards/Logic_Level_Bidirectional.pdf>. Acesso em: 16 ago. 2017.

- [24] NXP SEMICONDUCTORS. NXP Semiconductors | Automotive, Security, IoT. Disponível em: <<https://www.nxp.com/>>. Acesso em: 06 nov. 2017.
- [25] I2C BUS. **I2C Bus, Interface and Protocol**. Disponível em: <<http://i2c.info/>>. Acesso em: 17 ago. 2017.
- [26] SPARKFUN. **Sparkfun Education**. Disponível em: <<http://www.sparkfuneducation.com/index.html>>. Acesso em: 17 ago. 2017.
- [27] ALIEXPRESS. AliExpress.com - Comprar eletrônicos, produtos de moda, produtos para casa & jardim e automotivos vindos diretamente da China. Disponível em: <<https://pt.aliexpress.com/>>. Acesso em: 23 nov. 2017.
- [28] BANGGOOD. **Online Shopping for Cool Gadgets, RC helicopter & Quadcopter, Mobile Phone, Fashion at Banggood.com**. Disponível em: <<https://www.banggood.com/>>. Acesso em: 23 nov. 2017.
- [29] ELSEVIER. **Scopus - Document Search**. Disponível em: <<https://www.scopus.com/>>. Acesso em: 15 ago. 2017.
- [30] SPARKFUN. **Pull-up resistor**. Disponível em: <<https://learn.sparkfun.com/tutorials/pull-up-resistors>>. Acesso em: 17 ago. 2017.
- [31] SPARKFUN. **Bi-Directional Logic Level Hookup Guide**. Disponível em: <https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide?_ga=2.2962355.2052324461.1502145581-1316188455.1492522664>. Acesso em: 16 ago. 2017.
- [32] ARDUINO E CIA. **Arduino e Cia: Calculadora Online - Divisor de Tensão**. Disponível em: <<http://www.arduinoecia.com.br/p/calculador-divisor-de-tensao-function.html>>. Acesso em: 16 ago. 2017.
- [33] SPARKFUN. **Pro Micro & Fio V3 hookup Guide**. Disponível em: <<https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide>>. Acesso em: 17 ago. 2017.
- [34] SPARKFUN. **Pro-Micro-v11**. Disponível em: <https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/Pro-Micro-v11-5V_B.pdf>. Acesso em: 17 ago. 2017.
- [35] ORACLE CORPORATION. O que é o Java? Disponível em: <https://www.java.com/pt_BR/about/whatis_java.jsp?bucket_value=desktop-chrome62-

- windows10-64bit&in_query=no>. Acesso em: 24 nov. 2017.
- [36] ARDUINO. Arduino Reference Serial. Disponível em: <<https://www.arduino.cc/reference/en/language/functions/communication/serial/>>. Acesso em: 24 nov. 2017.
- [37] RXTX. RXTX. Disponível em: <http://rxtx.qbang.org/wiki/index.php/Main_Page>. Acesso em: 24 nov. 2017.
- [38] KLANG, R.; BAERVELDT, A.-J. A Low-cost and Low-weight Attitude Estimation System for an Autonomous Helicopter. Disponível em: <<http://www.geology.smu.edu/dpa-www/robo/balance/inertial.pdf>>. Acesso em: 28 nov. 2017.
- [39] BITTENCOURT DE ABREU, L. Estudo da Utilização de Filtros de Kalman para Auto Localização. Disponível em: <<https://repositorio-aberto.up.pt/bitstream/10216/57737/2/Texto%20integral.pdf>>. Acesso em: 28 nov. 2017.
- [40] ARDUINO E CIA. Acelerômetro e Giroscópio MPU6050. Disponível em: <<http://www.arduinoecia.com.br/2015/04/acelerometro-giroscopio-mpu-6050.html>>. Acesso em: 28 nov. 2017.
- [41] ELETRONICFANS. 1 Conjunto IIC I2C GY-521 MPU-6050 MPU6050 3 Eixos Giroscópio Analógico Sensores + 3 Eixos Acelerômetro Módulo Para Arduino Com Pinos 3-5 V DC. Disponível em: <https://pt.aliexpress.com/item/MPU-6050-3-Axis-gyroscope-accelerometer-module-3V-5V-compatible-ForArduino/2035920870.html?spm=a2g03.search0104.3.2.sBBBB1Q&ws_ab_test=searchweb0_0,searchweb201602_3_10152_10065_10151_10344_10068_10345_10342_10547_10343_1017>. Acesso em: 28 nov. 2017.
- [42] ALIEXPRESS. Disponível em: <<https://pt.aliexpress.com/item/High-quality-LSM9DS0-IMU-9DoF-High-precision-integrated-9-axis-attitude-sensor-I2C-SPI-Module-Free/32770547010.html?spm=a2g0s.9042311.0.0.oQ6IBH>>. Acesso em: 24 ago. 2017.
- [43] SPARKFUN. **Pro Micro & Fio V3 Hookup Guide - Installing: Windows.** Disponível em: <https://github.com/sparkfun/Arduino_Boards/archive/master.zip>. Acesso em: 18 ago. 2017.

- [44] ARDUINO. **Arduino - Software.** Disponível em: <<https://www.arduino.cc/en/Main/Software>>. Acesso em: 21 ago. 2017.
- [45] BOYLESTAD, R. L. **Análise de Circuitos.** São Paulo: Pearson Education, 2012.
- [46] ALPER, S.; RAHARINIRINA, S. Assistive Technology for Individuals with Disabilities: A Review and Synthesis of the Literature. **Journal of Special Education Technology**, 2006.
- [47] DE ANGELIS, V. **headClickMouse.ino.** Disponível em: <<https://www.dropbox.com/s/601jhsrxhzk0i95/headClickMouse.ino?dl=0>>. Acesso em: 18 ago. 2017.
- [48] WORLDCHIPS. Mini USB ATmega32U4 Pro Micro 5 V 16 MHz Módulo Board Para Arduino/ATMega 32U4 Leonardo Controlador Pro Micro Substituir Pro Mini em Circuitos integrados de Componentes eletrônicos & Suprimentos no AliExpress.com | Alibaba Group. Disponível em: <<https://pt.aliexpress.com/item/Pro-Micro-ATmega32U4-5V-16MHz-Replace-ATmega328-Pro-Mini/32649703074.html?spm=a2g0s.9042311.0.0.k8qaSd>>. Acesso em: 01 nov. 2017.
- [49] BANGGOOD. CJMCU LSM9DS0 IMU 9DoF High Precision Integrated 9-Axis Attitude Sensor SPI / I2C For Arduino Sale - Banggood.com sold out. Disponível em: <<https://www.banggood.com/CJMCU-LSM9DS0-IMU-9DoF-High-Precision-Integrated-9-Axis-Attitude-Sensor-SPI-I2C-For-Arduino-p-1101058.html?rmmds=myorder>>. Acesso em: 01 nov. 2017.
- [50] BANGGOOD. 5 Pcs 3.3V 5V TTL Bi-directional Logic Level Converter For Arduino Sale - Banggood.com. Disponível em: <https://www.banggood.com/5-Pcs-3_3V-5V-TTL-Bi-directional-Logic-Level-Converter-For-Arduino-p-951182.html?rmmds=myorder>. Acesso em: 01 nov. 2017.