

Projeto e Desenvolvimento de uma *Dataglove* de Baixo Custo para Aplicações em Realidade Virtual

Rabah Zeineddine, Luciano Silva

¹Faculdade de Computação e informática – Universidade Presbiteriana Mackenzie
São Paulo, SP – Brasil

41502779@mackenzista.com.br, luciano.silva@mackenzie.br

Abstract. *The present study aimed to develop a low-cost dataglove using an Arduino LilyPad board. In the first place, some models of datagloves and components that are fundamental to its operation were exposed. Afterward, the basic concepts of Datagloves and examples available and the average price were discussed, these datagloves can reach \$ 5,495. Finally, it is introduced the steps of the dataglove prototype development process and its results. It was concluded that, despite some limitations, it is possible to produce a low cost dataglove compared to the currently developed datagloves in the market, becoming more accessible to different populations.*

Resumo. *O presente estudo teve por objetivo desenvolver e apresentar uma luva de dados de baixo custo utilizando a placa de Arduino LilyPad. Em primeiro lugar, foram expostos alguns modelos de luvas e componentes fundamentais para seu funcionamento. Após foram colocados conceitos básicos de Datagloves e exemplos disponíveis no mercado e a média de preço, estas luvas podem chegar a \$ 5,495. Por fim, mostrou-se as etapas de realização do protótipo de luva e seus resultados. Concluiu-se que, apesar de algumas limitações, é possível a confecção da luva por um custo muito menor do que os encontrados no mercado, tornando seu uso mais acessível a diferentes populações.*

1. Introdução

Com o avanço da tecnologia e para tornar cada vez mais preciso e mais próximo da realidade a interação humano-computador surge a necessidade de novos dispositivos e hardwares mais potentes que ajudam a melhorar essa interação capturando e analisando cada vez mais dados precisos em tempo real.

Dataglove ou luva de dados, é uma luva fabricada com altas tecnologias que permitem capturar os gestos e os movimentos da mão do ser humano e transmitir esses dados para um computador, por exemplo.

Com a ajuda de softwares desenvolvidos especificamente para este tipo de aplicação dos dados capturados, é possível simular e apresentar alguns tipos de aplicações como do movimento da mão dentro da Realidade Virtual.

Hardwares pequenos de alta funcionalidade já estão disponíveis para o uso como a placa Arduino. O Arduino é um dispositivo relativamente pequeno de placa única que pode ser usado para objetos interativos independentes.

Junto ao Arduíno existe uma série de sensores e dispositivos plugáveis compatíveis com a placa para estender a sua capacidade a projetos cada vez mais complexos e completos.

Uma placa típica de Arduíno é composta por um controlador, algumas linhas de entrada e saída digital e analógica além da interface serial ou USB. A interação entre a placa e o computador, por exemplo, é feita através de alguns softwares desenvolvidos utilizando diferentes linguagens de programação.

Dentro deste contexto, este projeto apresenta os conceitos básicos de uma luva de dados e algumas aplicações dessa luva no mundo real como também uma comparação dos preços das luvas atuais.

Além da apresentação da luva, são apresentados os conceitos da placa Arduíno e alguns modelos existentes, especificamente apresentou-se a placa Arduíno *LilyPad* com seus detalhes e especificações, bem como da utilização da plataforma Unity3D para simular os dados capturados da luva.

Com esta placa de microcontrolador, o projeto propõe o desenvolvimento de uma luva de dados de baixo custo utilizando a placa de Arduíno *LilyPad* junto a alguns sensores externos para a captura de dados utilizando a plataforma Unity3D para a modulação e a simulação dos dados.

A organização deste trabalho está feita em seções, sendo que a seção 2 aborda conceitos básicos de *Datagloves*, onde foram demonstrados alguns exemplos.

Em seguida, apresentou-se na seção 3 a metodologia constando o detalhamento de todas as etapas para a produção do protótipo de luva desenvolvida.

Na seção 4, os resultados do projeto foram expostos, mostrando a simulação do movimento da luva de dados e os custos envolvidos para sua produção. Por fim, na seção 5, é apresentada a conclusão contendo uma síntese do estudo e sugestões de trabalhos futuros.

2. Referencial teórico

Esta seção apresenta os conceitos básicos da *dataglove* mostrando algumas aplicações na Realidade Virtual.

2.1. Conceitos básicos da *Datagloves*

A *Dataglove* ou também conhecida como *cyberglove* é um dispositivo de entrada de dados para a interação humano-computador fabricado no formato de uma luva vestível que captura o movimento da mão utilizando vários sensores tecnológicos [Büscher et al. 2012]



Figura 1. *Cyberglove III*. Fonte: <https://www.cyberglovesystems.com>

A mão do ser humano, especificamente, é uma maneira geral das pessoas interagirem com o ambiente real. Os gestos da mão são uma parte muito importante do corpo do ser humano a serem capturados e simulados num mundo virtual.

Com isso, pode-se ter um controle mais preciso e mais próximo da realidade usando uma simulação da mão capturada em mundo real e aplicada em mundo virtual como na Realidade Virtual ou Realidade Aumentada [Linn et al. 2017]

2.2. Exemplos da *Datagloves*

Há vários modelos da datagloves que já foram desenvolvidos com o objetivo de capturar os gestos e os movimentos da mão do ser humano e transmitir os dados capturados para o computador. Um exemplo deles, mostrado na figura 1, é a *CyberGlove III*.

A *Cyberglove III* possui até 22 sensores como também possui a tecnologia de transmissão de dados pelo wifi 802.11g, além disso a bateria dele dura 2 horas e captura por volta de 100 registro/seg. , ele possibilita a captura dos gestos da mão e a transmissão com umas tecnologias bem avançadas [Mardiyanto et al. 2017]

Um outro exemplo da *dataglove*, mostrado na figura 2, é a 5DT:



Figura 2. Dataglove 5DT. Fonte: <http://www.5dt.com>

A luva 5DT foi desenhada para satisfazer os requisitos modernos da captura de dados e animação profissional, oferecendo conforto e facilidade de uso. Ela interage com um computador via USB ou via WIFI, tendo 14 sensores para detecção do movimento da mão como também a bateria dura até 8 horas [O-larnnithipong and Barreto 2016]

Existe também outro conceito no modo de utilização de uma luva de dados ou *dataglove*, a *CyberGrasp*, por exemplo, uma luva com sistema inovador de retorno de força para os dedos e a mão, ou seja, a *CyberGrasp* permite que você sinta alguns objetos 3D gerados pelo computador em um mundo virtual, conforme mostrado na figura 3



Figura 3. CyberGrasp. Fonte: <https://www.cyberglovesystems.com>

A *CyberGrasp* acrescenta alguns resistores nos dedos a mais em comparação com a *CyberGlove III* para poder controlar a mão do ser humano. [O-larnnithipong and Barreto 2016]

2.3. Aplicações de *DataGloves* em Realidade Virtual

Os sistemas da Realidade Virtual dependem dos dispositivos de entrada de captura de dados de ambientes no mundo real e transmissão das informações adquiridas para um ambiente virtual. Este é um processo importante para providenciar o melhor dentro de um ambiente virtual.

Os gestos e os movimentos do ser humano são os dados mais capturados com os dispositivos de entrada de dados da Realidade Virtual [Oh and Whangbo 2017]

A *dataglove* é um dos dispositivos utilizado para a captura de gestos da mão como mostra a figura 4



Figura 4. *Dataglove* na realidade virtual. Fonte: <https://manus-vr.com>

A aplicação da *dataglove* na Realidade Virtual ajuda bastante no nível de precisão na captura dos gestos e movimentos da mão do ser humano, os dados são capturados em tempo real e transmitidos para um computador através de USB ou WIFI.

Após recebidos, os dados passam por um software no qual é feita a simulação para produzir o modelo de uma mão virtual dentro do ambiente virtual [Tham et al. 2018]



Figura 5. *Simulação de CyberGlove II*. Fonte <http://www.cyberglovesystems.com>

A figura 5 mostra um modelo computacional utilizando *CyberGlove II* como dispositivo de captura de gestos onde percebe-se que a mão foi modulada perfeitamente, este modelo pode ser utilizado dentro do mundo virtual.

2.4. Comparação de Preços de *Datagloves*

As *datagloves* existentes no mercado e alguns representados acima, tem um preço alto que varia entre 500 \$ e chega até 5,495 \$, ou seja, um preço bem elevado para ser disponível para qualquer pessoa.

A figura 6 mostra alguns preços de alguns dataglove 5DTs:

ITEM	DESCRIPTION	PRICE
DG05UR	5DT Data Glove 5 Ultra (Right)	\$995
DG05UL	5DT Data Glove 5 Ultra (Left)	\$995
DG14UR	5DT Data Glove 14 Ultra (Right)	\$5,495
DG14UL	5DT Data Glove 14 Ultra (Left)	\$5,495
DGUWK	5DT Data Glove Wireless Kit (for 2 gloves)	\$1,495
DGUDMOB	MotionBuilder Driver for 5DT Data Gloves	\$495

Figura 6. *5DT CyberGlove Pricing*. Fonte: <http://www.5dt.com>

O preço da luva pode variar, dependendo de diversos aspectos como a quantidade de sensores e qualidade deles, o material utilizado na fabricação da luva como também alguns dispositivos de WiFi e Bluetooth que podem ser utilizados para a transmissão de dados [Pereira et al. 2013]

2.5. Arduíno

O Arduíno é uma plataforma eletrônica aberta baseada na facilidade de uso de hardware e de software. As placas de Arduíno são capazes de ler entradas como luz de sensores, botões, entre outros.

Ademais, é responsável por transformar isso em uma saída de dados, ou seja, dados capturados transformados em dados computacionais.

A figura 7 mostra uma placa de Arduíno UNO:



Figura 7. *Arduíno UNO USB*. Fonte: <https://www.arduino.cc>

A ideia de criar o Arduíno foi disponibilizar uma plataforma para providenciar um hardware de desenvolvimento aberto para criação de dispositivos e projetos, integrando vários tipos de sensores e dispositivos plugáveis compatíveis.

Após o crescimento do Arduíno no mercado, vários dispositivos foram fabricados para servir a algum propósito específico, como *Wireless*, sensores, *Bluetooth*, entre outros. Arduíno tem a facilidade de aprender linguagem e bibliotecas baseadas em C++ como também um IDE para a própria interface de programação.

O Arduíno é considerado como plataforma de hardwares independentes, que pode operar nos sistemas operacionais Windows, Linux ou MAC.

2.6. LilyPad

Há vários modelos de placas de Arduínos fabricadas até o momento, como Arduíno Uno, Arduíno Duemilanove, Arduíno Diecimila e várias outras placas com especificações diferentes de capacidade e compatibilidade de hardware.

O LilyPad [Buechley and Eisenberg 2008] é um tipo de placa de Arduíno que foi desenhada para tecnologias têxteis e projetos vestíveis de tamanho pequeno.

A figura 8 mostra uma placa Arduíno LilyPad:

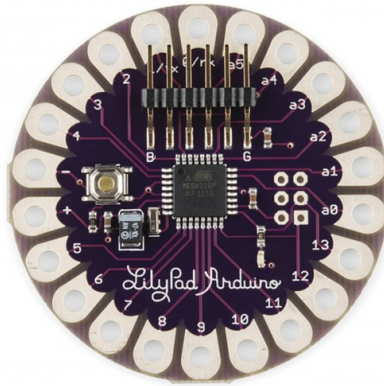


Figura 8. Arduino LilyPad. Fonte: <https://www.arduino.cc>

A placa de Arduino *LilyPad* é baseada em ATmega168V, uma versão de baixo consumo de energia de ATmega168 ou ATmega328V. ATmega168 é um microcontrolador AVR baseado na arquitetura RISC de 8-bit. ATmega168 combina com 16KB ISP memória Flash com a capacidade de ler enquanto escreve, 512B EEPROM, 1KB SRAM, 23 linhas de entrada e saída, 32 registradores entre outras especificações técnicas.

Com todas as especificações do *LilyPad*, ele apresenta um alto desempenho como também um consumo muito baixo de energia.

Uma placa de Arduino com um tamanho pequeno igual *LilyPad* e com um desempenho razoavelmente alto, pode ser utilizada em projetos vestíveis e robóticos como uma luva de dados e outros projetos que necessitam uma placa pequena onde pode ser portátil e oculta.

3. Metodologia

Para entender melhor como as placas Arduino funcionam e como é possível conectar, programar e montar as peças para desenvolver a luva, foi preciso primeiramente estudar sobre a plataforma Arduino que permite a interação entre o computador e as placas Arduino.

Com esta plataforma oferecida pela própria fábrica de Arduino teremos como programar e customizar o código executado na placa para algum tipo de funcionamento como, por exemplo, interação entre a placa e algum sensor externo.

Alguns experimentos e testes básicos foram feitos na placa Arduino UNO junto com a plataforma Arduino como apresentado na figura 9, capaz de acender e apagar o LED do próprio Arduino com intervalo de 1 segundo entre cada ação.


```

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

```

Figura 9. Blink code

Foi realizado um estudo sobre quais dispositivos serão necessários para o desenvolvimento da luva de dados e com isso foi identificado que será necessário utilizar sensores flex que nos permitem calcular o ângulo de inclinação de cada dedo utilizando um algoritmo que converte a resistência de cada sensor em um ângulo.

Já a placa de Arduino *Lilypad* permite o funcionamento de uma placa Arduino em dimensões menores e vestíveis como mencionado na seção 2.6.

São necessários resistores de 47 KOHM utilizados junto com os sensores flex, acelerômetro e giroscópio para conseguirmos calcular a rotação e o movimento da luva em tempo real e um módulo *XBEE* para transmitir o sinal sem fio.

Com os dispositivos identificados, alguns trabalhos foram feitos sobre os sensores flex, resistores e a placa Arduino UNO, onde foi conectado cada sensor com um resistor a uma porta analógica da placa Arduino.

A conexão foi feita em série como a figura 10 mostra.

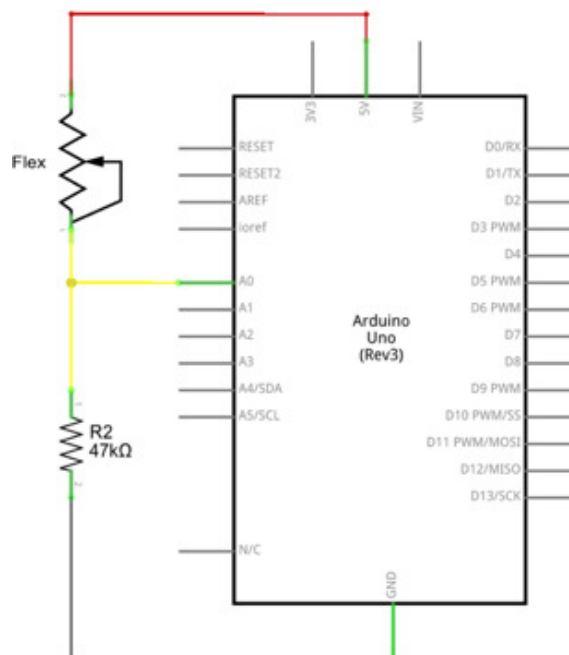


Figura 10. Fonte: <https://learn.sparkfun.com>

Foi utilizado um código que recupera os dados da porta analógica onde o sensor

flex está conectado e faz um cálculo para identificar o ângulo da inclinação do sensor, assim teria como saber os gestos de cada dedo.

A figura 11 mostra o código utilizado.

```

const int FLEX_PIN = A0; // Pin connected to voltage divider output
// Measure the voltage at 5V and the actual resistance of your
// 47k resistor, and enter them below:
const float VCC = 4.98; // Measured voltage of Arduino 5V line
const float R_DIV = 47500.0; // Measured resistance of 3.3k resistor
// Upload the code, then try to adjust these values to more
// accurately calculate bend degree.
const float STRAIGHT_RESISTANCE = 37300.0; // resistance when straight
const float BEND_RESISTANCE = 90000.0; // resistance at 90 deg
void setup() {
  Serial.begin(9600);
  pinMode(FLEX_PIN, INPUT);
}

void loop()
{
  // Read the ADC, and calculate voltage and resistance from it
  int flexADC = analogRead(FLEX_PIN);
  float flexV = flexADC * VCC / 1023.0;
  float flexR = R_DIV * (VCC / flexV - 1.0);
  Serial.println("Resistance: " + String(flexR) + " ohms");

  // Use the calculated resistance to estimate the sensor's
  // bend angle:
  float angle = map(flexR, STRAIGHT_RESISTANCE, BEND_RESISTANCE,
    0, 90.0);
  Serial.println("Bend: " + String(angle) + " degrees");
  Serial.println();

  delay(500);
}

```

Figura 11. Código utilizado para leitura de dados do sensor flex

O código foi modificado para suportar 5 sensores flex, assim teria como capturar os gestos dos 5 dedos de uma mão como mostra a figura 12 abaixo.

```

void setup()
{
  Serial.begin(9600);
  pinMode(FLEX_PIN_0, INPUT);
  pinMode(FLEX_PIN_1, INPUT);
  pinMode(FLEX_PIN_2, INPUT);
  pinMode(FLEX_PIN_3, INPUT);
  pinMode(FLEX_PIN_4, INPUT);
}

void loop()
{
  // Read the ADC, and calculate voltage and resistance from it
  int flexADC_0 = analogRead(FLEX_PIN_0);
  int flexADC_1 = analogRead(FLEX_PIN_1);
  int flexADC_2 = analogRead(FLEX_PIN_2);
  int flexADC_3 = analogRead(FLEX_PIN_3);
  int flexADC_4 = analogRead(FLEX_PIN_4);

  float flexV_0 = flexADC_0 * VCC / 1023.0;
  float flexV_1 = flexADC_1 * VCC / 1023.0;
  float flexV_2 = flexADC_2 * VCC / 1023.0;
  float flexV_3 = flexADC_3 * VCC / 1023.0;
  float flexV_4 = flexADC_4 * VCC / 1023.0;

  float flexR_0 = R_DIV * (VCC / flexV_0 - 1.0);
  float flexR_1 = R_DIV * (VCC / flexV_1 - 1.0);
  float flexR_2 = R_DIV * (VCC / flexV_2 - 1.0);
  float flexR_3 = R_DIV * (VCC / flexV_3 - 1.0);
  float flexR_4 = R_DIV * (VCC / flexV_4 - 1.0);

  // Use the calculated resistance to estimate the sensor's
  // bend angle:
  float angle_0 = map(flexR_0, STRAIGHT_RESISTANCE, BEND_RESISTANCE,
    0, 90.0);
  float angle_1 = map(flexR_1, STRAIGHT_RESISTANCE, BEND_RESISTANCE,
    0, 90.0);
  float angle_2 = map(flexR_2, STRAIGHT_RESISTANCE, BEND_RESISTANCE,
    0, 90.0);
  float angle_3 = map(flexR_3, STRAIGHT_RESISTANCE, BEND_RESISTANCE,
    0, 90.0);
  float angle_4 = map(flexR_4, STRAIGHT_RESISTANCE, BEND_RESISTANCE,
    0, 90.0);
  Serial.println("Bend 0 : " + String(angle_0) + " degrees");
  Serial.println("Bend 1 : " + String(angle_1) + " degrees");
  Serial.println("Bend 2 : " + String(angle_2) + " degrees");
  Serial.println("Bend 3 : " + String(angle_3) + " degrees");
  Serial.println("Bend 4 : " + String(angle_4) + " degrees");
  Serial.println();
}

```

Figura 12. Código para capturar os gestos dos dedos

A figura 13 mostra o protótipo feito para capturar os gestos dos 5 sensores junto com o Arduino UNO.

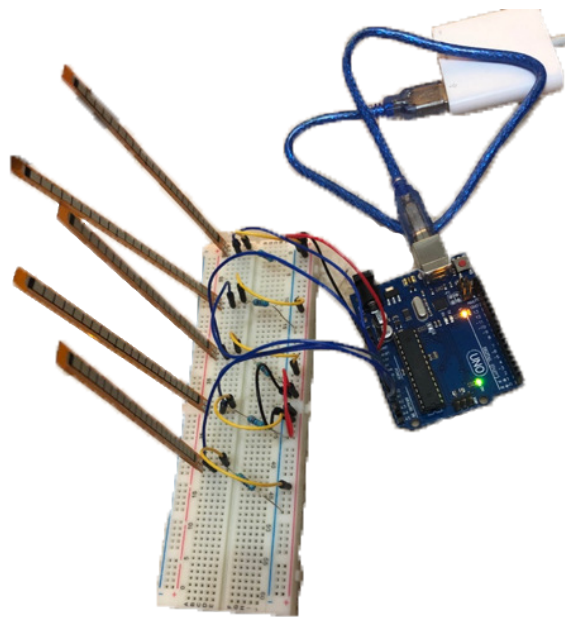


Figura 13. Protótipo de sensores flex com Arduino UNO.

Como resultado da conexão feita dos sensores com o Arduino UNO como mostra a figura 13 e o código desenvolvido na figura 12 foi possível capturar e calcular o ângulo de cada sensor como mostra a figura 14.

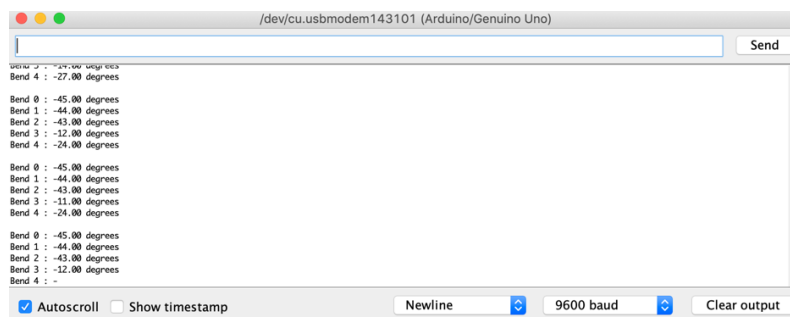


Figura 14. Ângulos dos sensores capturados

Para ter mais dados como a rotação e aceleração da luva, foi escolhido e estudado o dispositivo MPU-6050 que permite ter ambos os dados, portanto, foi realizado alguns testes e implementação com o Arduino UNO como a figura 15 mostra.

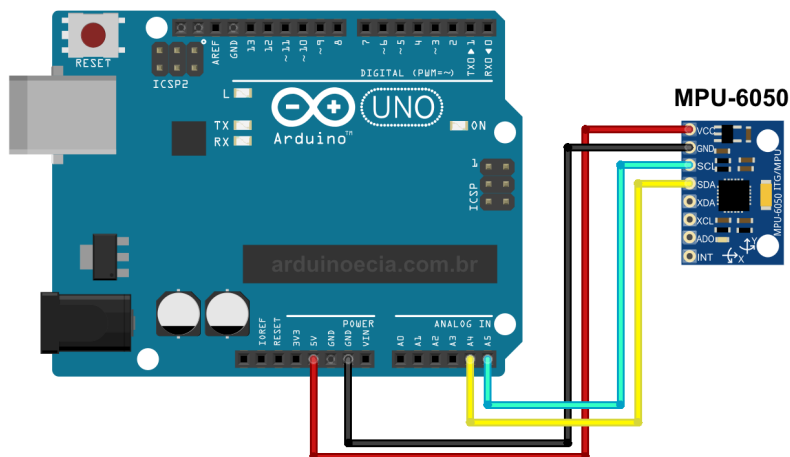


Figura 15. Circuito MPU-6050. Fonte: <https://www.arduinoecia.com.br>

Com os testes realizados dos sensores flex e o dispositivo MPU-6050 de giroscópio e acelerômetro foi feita a construção da luva de dados substituindo o Arduino UNO pelo Arduino Lilypad por motivos de dimensões da placa.

Além disso, também foi estudado sobre o dispositivo XBEE que permite a transmissão de dados sem fio necessitando de dois dispositivos, um utilizado na própria luva como remetente e outro utilizado no computador como receptor.

Desta forma, foi possível o envio dos dados capturados dos sensores flex, giroscópio e acelerômetro para o computador.

A figura 16 mostra o diagrama da luva de dados desenvolvido.

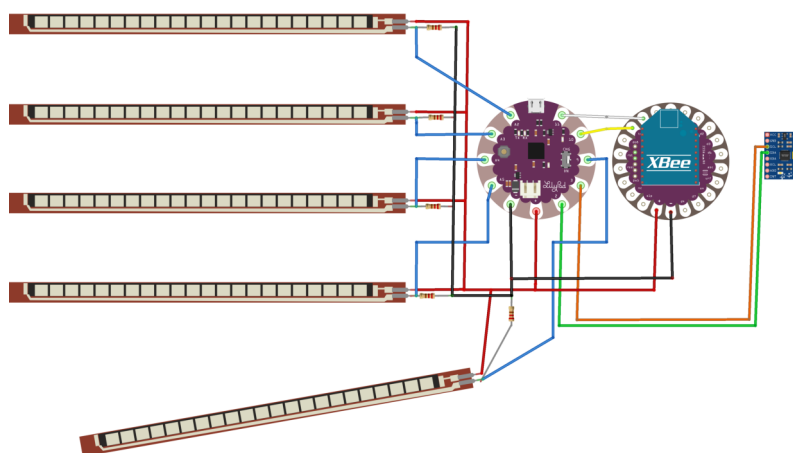


Figura 16. Schema da luva de dados

Para simular os dados capturados da luva de dados, foi preciso desenvolver uma aplicação no ambiente Unity3D que consegue ler os dados recebido via XBEE e atribuir a um objeto 3d desenvolvido que represente a mão.

A figura 17 mostra a aplicação desenvolvida no ambiente Unity3D para simular a

luva.

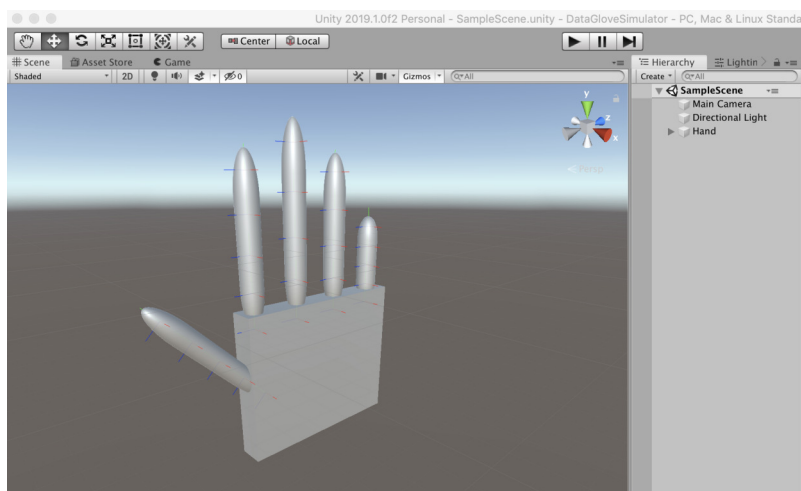


Figura 17. Aplicação desenvolvida em Unity3D para simulação

4. Resultados

Os testes preliminares foram feitos na placa Arduino UNO pela facilidade de uso e a possibilidade de conexões sem a necessidade de solda permanente. Após ter sido encontrada a solução correta, esta foi aplicada à placa de Arduino *Lilypad*

Foi criada uma luva de dados capaz de capturar os dados necessários da mão e enviá-los para um computador sem fio para serem analisados e simular a mão humana dentro de várias aplicações.

A figura 18 mostra a luva de dados desenvolvida.

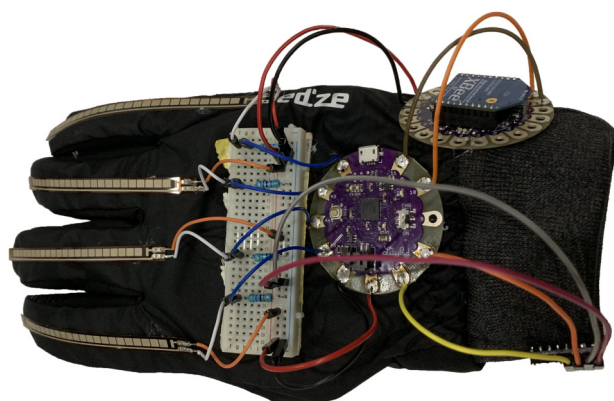


Figura 18. Luva de dados desenvolvida

Junto com a luva desenvolvida e a aplicação desenvolvida no ambiente Unity3d (figura 17), foi possível ter uma simulação da luva com movimento e simulado na aplicação como as figuras 19 e 20 mostram.



Figura 19. Movimento com a luva de dados

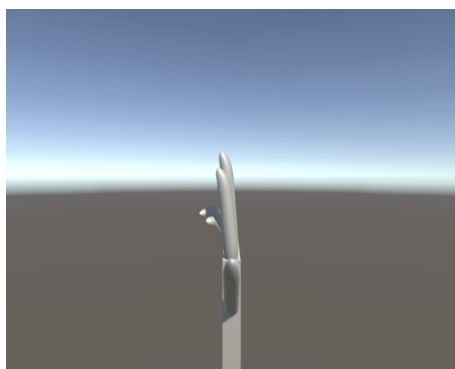


Figura 20. Simulação do movimento da luva de dados

A produção desta luva teve um custo de \$ 120, valor menor do que os preços praticados pelo mercado como mostra a figura 6.

5. Conclusões e Trabalhos Futuros

O estudo mostrou que a principal vantagem encontrada na luva criada é seu baixo custo, para a produção de nosso protótipo foram gastos em média 120 dólares. Comparado a luvas disponíveis no mercado que podem chegar a 5,495 dólares.

Esta é uma diferença de quase 50 vezes menor, abrindo possibilidade para que este tipo de produto seja acessível a um número maior de pessoas.

Já a principal desvantagem é que a luva produzida não é tão funcional quanto às profissionais comercializadas no mercado porque tem poucos sensores que permitem ao usuário fazer o mínimo.

Entretanto, ainda assim poderia ser indicada para uso pessoal ou comercial, desde que observadas as necessidades do uso. Pois é possível utilizá-la para jogos virtuais, na Realidade Virtual, tradução da linguagem em libras ou quaisquer aplicações que dependam de um dispositivo externo para controle.

Portanto, com esta luva de dados desenvolvida, os objetivos do projeto foram atingidos.

Dentre sugestões para pesquisas futuras, a primeira seria verificar a compatibilidade de adicionar alguns sensores na parte dos dedos da luva para garantir o movimento de distância entre eles, o que aumentaria a precisão dos dados capturados e tornaria a simulação mais próxima da realidade.

Outra questão de interesse, seria a possibilidade da criação de um modelo 3d da luva utilizando a impressora 3d, que facilitaria a montagem da luva definindo as posições corretas de cada sensor utilizado, garantindo as mesmas funcionalidades para todos que desenvolvessem a luva como foi criada neste trabalho.

Por fim, sugere-se a criação de uma placa específica para esta luva similar à placa Arduíno *Lilypad* utilizada, mas com dimensões menores para ocupar menos espaço na luva e operar da mesma maneira.

Referências

- Buechley, L. and Eisenberg, M. (2008). The lilypad arduino: Toward wearable engineering for everyone. *IEEE Pervasive Computing*, 7(2):12–15.
- Büscher, G., Kõiva, R., Schürmann, C., Haschke, R., and Ritter, H. J. (2012). Tactile dataglove with fabric-based sensors. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 204–209.
- Linn, T., Jwaid, A., and Clark, S. (2017). Smart glove for visually impaired. In *2017 Computing Conference*, pages 1323–1329.
- Mardiyanto, R., Utomo, M. F. R., Purwanto, D., and Suryoatmojo, H. (2017). Development of hand gesture recognition sensor based on accelerometer and gyroscope for controlling arm of underwater remotely operated robot. In *2017 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pages 329–333.
- O-larnnithipong, N. and Barreto, A. (2016). Gyroscope drift correction algorithm for inertial measurement unit used in hand motion tracking. In *2016 IEEE SENSORS*, pages 1–3.
- Oh, S. H. and Whangbo, T. K. (2017). A study on the effective interaction method to improve the presence in social virtual reality game. In *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, pages 1–2.

- Pereira, L. C., Aroca, R. V., and Dantas, R. R. (2013). Flexdglove: A low cost dataglove with virtual hand simulator for virtual reality applications. In *2013 XV Symposium on Virtual and Augmented Reality*, pages 260–263.
- Tham, J., Duin, A. H., Gee, L., Ernst, N., Abdelqader, B., and McGrath, M. (2018). Understanding virtual reality: Presence, embodiment, and professional practice. *IEEE Transactions on Professional Communication*, 61(2):178–195.