

# Development of a sensor glove for finger tracking and physiological measurement to Virtual Reality applications

## Desenvolvimento de uma luva com rastreamento de dedos e monitoramento fisiológico para aplicações de Realidade Virtual

Samuel Antunes Vieira  
Universidade de Passo Fundo  
Passo Fundo, RS, Brazil  
158820@upf.br

Rafael Rieder  
Universidade de Passo Fundo  
Passo Fundo, RS, Brazil  
rieder@upf.br

### ABSTRACT

VR Wear is a low-cost wearable prototype supporting finger tracking, developed for immersive virtual environments created with the Unity game engine. This device also allows connecting physiological monitoring sensors, providing a versatile tool useful in different interactive-context projects. This paper describes related work, background concepts, the VR Wear device project and development, and shows preliminary tests of the equipment.

### RESUMO

VR Wear é um protótipo vestível no qual se busca provar o conceito de rastreamento de dedos, de baixo custo, para ambientes virtuais imersivos projetados na game engine Unity. O dispositivo também permite o acoplamento de sensores de monitoramento fisiológico, fornecendo um acessório multifuncional capaz de ser utilizado em aplicações interativas de diferentes propósitos. Este artigo descreve projetos similares, os conceitos relacionados, o projeto e o desenvolvimento do VR Wear, e apresenta testes iniciais do equipamento.

### CCS CONCEPTS

• **Computing methodologies** → **Virtual reality**; • **Human-centered computing** → **Haptic devices**; • **Hardware** → **Haptic devices**.

### KEYWORDS

finger tracking, glove, Unity, Virtual Reality, wearable

### PALAVRAS-CHAVE

luva, rastreamento de dedos, Realidade Virtual, Unity, vestível

#### ACM Reference Format:

Samuel Antunes Vieira and Rafael Rieder. 2021. Development of a sensor glove for finger tracking and physiological measurement to Virtual Reality applications Desenvolvimento de uma luva com rastreamento de dedos

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*SVR '21, October 18–21, 2021, Gramado, RS, Brazil*

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06... \$15.00

<https://doi.org/10.1145/1122445.1122456>

e monitoramento fisiológico para aplicações de Realidade Virtual. In *SVR '21: 23rd Symposium on Virtual and Augmented Reality, October 18–21, 2021, Gramado, RS, Brazil*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/1122445.1122456>

### 1 INTRODUÇÃO

De acordo com Seidel e Chatelier [21], a Realidade Virtual (RV) pode ser definida como a imersão humana em um ambiente tridimensional sintético, gerado automaticamente e em tempo real. Tecnologias como head-mounted displays (HMDs) e rastreadores de movimento dão suporte à visualização e a interação nesses ambientes, possibilitando uma experiência imersiva aos usuários.

Uma das formas de explorar ambientes virtuais imersivos é por meio de controles, que permitem a interação direta através de técnicas de mão virtual [15], ou o deslocamento na cena utilizando metáforas de locomoção, como teleporte [8]. Novas tecnologias de RV também têm sido criadas, como controles hápticos portáteis e multifuncionais para tarefas de manipulação [5], e vestíveis para manipulação natural de objetos tangíveis [7].

Por meio de vestíveis, uma simulação virtual pode ficar mais natural, principalmente por não depender de fios conectados ao computador. Segundo Mann [17], um vestível facilita a interação humano-computador, pois é projetado para estar próximo ao corpo e sempre acessível. Ele está incluído no espaço pessoal do usuário, é controlado por este, e tem constância operacional.

Dispositivos vestíveis são fabricados para serem leves e darem liberdade de movimentos, podendo fornecer tanto feedback tátil quanto rastreamento em tempo real [20]. Isso favorece sua utilização em simulações de feedback tátil e de força sobre objetos virtuais. Além disso, podem ser utilizados para coleta de dados fisiológicos e interface com usuário [2], por um preço reduzido.

Uma luva, por exemplo, pode abranger diversos fatores, como sensação de toque, localização e rastreamento dos dedos, mãos e braços no ambiente de simulação, reconhecimento de gestos, e feedback de força. Nesse contexto, o uso de vestíveis como luvas são importantes, pois permitem o mapeamento das mãos ao longo do processo interativo.

Com isto em mente, este artigo apresenta o **VR Wear**, uma luva com rastreamento de dedos em tempo real, integrada a um vestível de monitoramento fisiológico, de custo acessível, que possibilita acompanhamento do usuário em aplicações de RV.

Para tanto, esse trabalho está assim organizado: a Seção 2 apresenta trabalhos relacionados, a Seção 3 mostra os componentes utilizados, a Seção 4 apresenta o protótipo criado, a Seção 5 relata

os resultados e discussões, e a Seção 6 finaliza com conclusões e trabalhos futuros.

## 2 TRABALHOS RELACIONADOS

Kim *et al.* [13] produziram um vestível háptico com rastreamento utilizando Leap Motion e feedback de força, para o pulso, o polegar e o indicador, provendo compatibilidade entre estes diferentes sistemas. Eles criaram uma interface prática e intuitiva, na qual o usuário pode interagir com objetos virtuais abstratos, como um player de música e botões, ou utilizar da própria mão como objeto, permitindo o disparo de uma bala da ponta do indicador, conforme o movimento do polegar.

Kervegant *et al.* [12] integraram Leap Motion com Ultrahaptics, permitindo rastreamento e feedback para as mãos em um sistema de RA. Essa solução permitiu ver, tocar e sentir um objeto virtual como se fosse real, comprometendo, porém, a portabilidade, com uma área limitada de uso, e aumentando a complexidade do sistema.

Micelli *et al.* [19] desenvolveram um sistema de rastreamento baseado em unidade de medição inercial (IMU) para braços e pulsos com comunicação sem fio, permitindo maior liberdade sobre menos uso computacional.

Wilk *et al.* [25] indicam que sistemas de rastreamento baseados em câmeras não só limitam a liberdade de movimento de usuários, como também não são adequados para rastreamento preciso de vários membros. Quando sobrepostos em frente à câmera, se tornam inabilitados de serem rastreados e são sensíveis à luz, não havendo alternativa de calibração. Com base nisso, utilizaram a VR Glove [6] integrada a um HMD com algoritmos de reconhecimento de gestos para realizar um rastreamento preciso e confiável.

Chen *et al.* [4] desenvolveram uma solução que não utiliza IMUs, mas sensores eletromagnéticos e eletroímãs integrados num circuito impresso, em conjunto com filtro passa-banda, permitindo o rastreamento das pontas dos dedos em três dimensões com interferência quase nula. Para tanto, utilizaram o processo de trilateração e de algoritmos eficientes aplicados sobre os dados coletados, aplicando fórmulas simplificadas para que não fossem necessários componentes de alto processamento, reduzindo também o custo do sistema. A placa desenvolvida permite o rastreamento de até cinco dedos, usando diferentes frequências para cada sensor.

Como diferencial a estas abordagens, este trabalho apresenta um projeto com compatibilidade entre componentes, com rastreamento preciso da mão e dos dedos com baixa latência, resultando em um protótipo vestível de custo acessível. Além disso, a luva dispõe de sensores fisiológicos (e permite acoplar outros), os quais podem ser utilizados para monitorar sinais vitais e funções corporais básicas do usuário durante o processo interativo.

## 3 COMPONENTES

Para o desenvolvimento do projeto, utilizou-se uma luva de tecido, onde os seguintes dispositivos estarão conectados com jumpers: microcontrolador Arduino Lilypad, sensor de oximetria de pulso e monitor de frequência cardíaca, e IMUs. O envio dos dados para a aplicação é realizado por bluetooth. O software utilizado para a comunicação entre os componentes foi escrito em linguagem C, com auxílio da biblioteca avr-libc e compilado com GCC (GNU Compiler Collection).

A placa Atmega328P é responsável pelo gerenciamento dos sensores, processamento de dados e é o componente que desempenha as ações a serem tomadas pelo sistema. Ela foi desenvolvida originalmente por Leah Buechley e SparkFun Electronics, sendo baseada nos modelos ATmega168 e Atmega328V [3].

O sensor MAX30100 é responsável pelo monitoramento simultâneo da frequência cardíaca e da pressão arterial dos usuários [18]. Os batimentos e a pressão arterial são obtidos por LEDs e fotodetectores, que permitem a detecção do momento da pulsação na circulação sanguínea.

Para a mensuração do movimento e do gasto calórico do usuário, e constante rastreamento da posição de seus dedos, optou-se pelo uso de duas diferentes IMUs. Para o primeiro requisito, optou-se pelo sensor MPU6050, uma IMU de baixo custo e precisão escalável em seis graus de liberdade (DOF) [9], cuja função principal será a aquisição de dados para a contagem de passos (pedômetro). Para o rastreamento da orientação espacial dos dedos, foi escolhido o sensor MPU9250 [10], que atende requisitos importantes como comunicação serial e 9DOF (acelerômetro, giroscópio e magnetômetro).

Como fonte de alimentação, definiu-se pelo uso de baterias tipo moeda ou LiPo (Lithium Polymer), de baixo custo e baixo consumo energético, aplicável a um dispositivo vestível e que atende a uma tensão entre 3V e 5V.

Para a comunicação serial, optou-se pelo uso do protocolo  $I^2C$ , na qual é realizado o envio de bytes de informações entre mestre (microcontrolador) e diversos escravos (sensores) numa mesma linha de comunicação. Como existe apenas um microcontrolador gerenciando diversos sensores com o mesmo endereço  $I^2C$ , adicionou-se um dispositivo de multiplexação, o TCA9548A  $I^2C$  Multiplexer, capaz de selecionar até 8 saídas de comunicação.

Para a comunicação entre o vestível e uma aplicação de RV, optou-se pela utilização do protocolo Bluetooth para comunicação sem fio. Para implementação desta combinação, o dispositivo utilizado (em conjunto com Arduino Lilypad), foi o módulo Bluetooth HC-05 [11].

Para a construção do **VR Wear**, investiu-se na compra dos componentes citados, com um custo total de, aproximadamente, R\$ 600,00, sem inclusão de lucros (com base de preços em Dezembro/2020). Esse valor demonstra o baixo custo da implementação do projeto, levando em consideração tecnologias similares [6, 13].

## 4 PROJETO E DESENVOLVIMENTO

A Figura 1 representa o diagrama de fluxo de dados do sistema, possibilitando uma melhor visualização da comunicação entre o vestível proposto e uma simulação em RV. Os sensores de monitoramento fisiológico e o multiplexador  $I^2C$  são conectados diretamente ao microcontrolador (pinos SDA e SCL), enquanto que as IMUs são conectadas ao multiplexador. Após a coleta de dados dos sensores, a placa Lilypad realiza o processamento destes dados e envia-os via USART (Universal Synchronous Asynchronous Receiver Transmitter) ao módulo Bluetooth, que retransmite esses sinais à um computador com suporte Bluetooth, que, em seguida, realiza as simulações em 3D por meio da Unity game engine.

A Figura 2 demonstra a estrutura do vestível e a distribuição dos componentes pela mão. As IMUs são fixadas nos dedos, o microcontrolador fica sobre o pulso, e multiplexador  $I^2C$  e módulo bluetooth são agregados às costas da mão. São utilizados jumpers

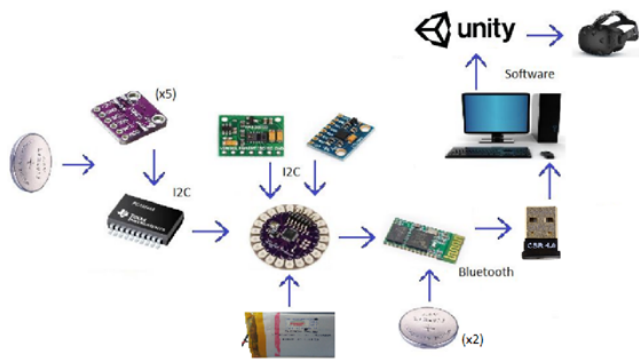


Figura 1: Fluxo de dados e comunicação entre componentes do protótipo.

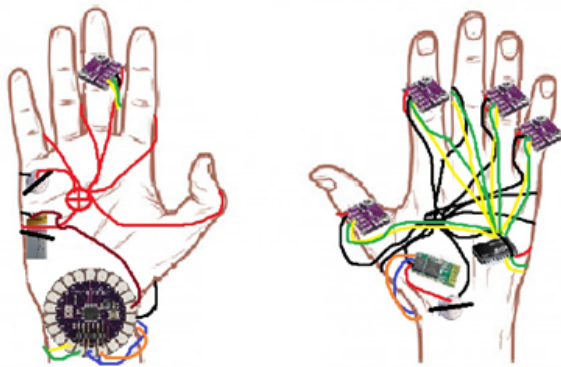


Figura 2: Estrutura e distribuição dos componentes no protótipo (mão direita).

para a conexão entre os componentes, sendo que linhas vermelhas representam fonte de alimentação de linhas pretas, aterramento. As próximas subseções descrevem materiais e métodos aplicados.

### 4.1 Unity

A Unity é uma ferramenta multiplataforma que possibilita a criação de games e simulações em 2D, 3D, RV e RA, gratuita para estudantes e compatível com a maior parte dos HMDs presentes no mercado [23]. Devido a isto, optou-se em desenvolver uma integração que permite ao vestível proposto se comunicar com uma aplicação de RV criada nesta ferramenta, permitindo a interação entre o usuário e o produto com tarefas de um ambiente virtual.

### 4.2 Esquemático e Comunicações

A Figura 3 representa o esquemático dos componentes integrados à luva. São utilizados resistores de pull-up de  $1K\Omega$  para garantir a estabilidade da comunicação  $I^2C$ .

Os dados são enviados via Bluetooth (conectado à USART do microcontrolador) e recebidos por um dispositivo USB receptor Bluetooth (CSR 4.0), conectado a um computador pessoal. Este transfere os dados para uma porta serial virtual [22], os quais são coletados e transformados em orientação 3D.

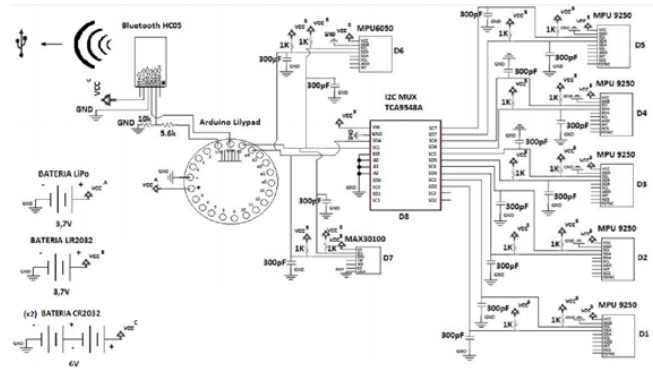


Figura 3: Esquemático do protótipo VR Wear.

### 4.3 Software

O microcontrolador foi programado com auxílio da IDE Atmel Studio em conjunto com a biblioteca avr-libc, e do programa de carregamento AVR Burn-O-Mat. Os dados processados são então enviados via Bluetooth à simulação em RV projetada na Unity engine, com scripts na linguagem C#.

Testes foram realizados com o microcontrolador Arduino Uno, enquanto que o carregamento do programa final foi feito por meio do mesmo como ISP (In-system programming), tendo a placa Arduino Lilypad como destino. Os testes com a IMU foram visualizados com auxílio das ferramentas serial monitor e serial plotter da Arduino IDE, que permitem a projeção de múltiplos dados em uma mesma linha temporal. Além da comunicação entre componentes, o software desenvolvido buscou configurar os sensores utilizados e coletar dados, aplicando filtros e conversões.

### 4.4 Configuração

A configuração dos componentes inicia-se pelo microcontrolador, que delimita as taxas de transmissão de dados. O modo de operação  $I^2C$  utilizado foi o modo rápido (fast mode), de 400kHz. Definiu-se também o baudrate em 38400bps para a comunicação USART.

Para a configuração do MPU9250, considerou-se três conceitos fundamentais: ADC word length, que representa a quantidade de bits (escala digital) em que o valor analógico é convertido; fator de escala de sensibilidade, que indica o quanto cada unidade binária (bit) representará em relação a unidade de medida especificada pelo fabricante; alcance em máxima escala, que indica o valor máximo a ser representado na unidade de medida especificada pelo fabricante.

A partir destas especificações, e considerando os fatores físicos do projeto, optou-se pela escolha de alcance em máxima escala de aceleração equivalente a  $\pm 4g$  (na busca de balancear movimentos rápidos e lentos realizados pelos braços e dedos), e de velocidade angular de  $\pm 250^\circ/s$  (já que os dedos possuem giro limitado às articulações humanas). Quanto ao magnetômetro, para uma maior precisão, são utilizados 16 bits para representação dos dados.

### 4.5 Filtros e Calibração

O sensor 9DOF conta com um filtro passa baixa integrado, selecionável para acelerômetro e giroscópio. Com isso, foi selecionado o filtro com banda de 10Hz, melhorando o sinal de ambos, porém,

diminuindo a frequência de atualização de dados de 8Khz do giroscópio para 1Khz e de 4Khz do acelerômetro também para 1Khz.

Após a seleção das configurações iniciais e do filtro passa baixa, executa-se um segmento de código em que são aplicadas calibrações nos registradores dos sensores e filtros nos dados recebidos.

Para calibração do giroscópio e do acelerômetro, são efetuadas cálculos de médias de valores, úteis para calcular o deslocamento (offset) e bias das próximas leituras. Como os valores de bias devem ser positivos, e sabendo-se que os valores lidos contém números negativos, ajustam-se as médias para tal fator.

Construídos os biases do giroscópio, enviam-se estes valores aos registradores de deslocamento que são utilizados para remover o bias DC (Direct Current) dos dados de saída do giroscópio antes que os mesmos sejam lidos. Em seguida, o bias é dividido pelo fator de sensibilidade configurado e esse resultado é armazenado para cálculos de ajuste de medidas.

Quanto ao acelerômetro, são coletados dados de deslocamento dos registradores, que são subtraídos dos biases calculados, e esse deslocamento é ajustado e reenviado aos registradores.

A calibração do magnetômetro inicia-se com a leitura de registradores que indicam valores de sensibilidade, utilizados para calcular biases e escalas de cada eixo do magnetômetro. São coletadas 1500 amostras em cada eixo (x, y e z), e realizadas médias denominadas no projeto como heavyiron e softiron.

Essa calibração é fundamental para o bom funcionamento do magnetômetro, já que o mesmo não implementa correções nos registradores internos. No momento inicial do programa, é importante que o usuário realize operações rotativas com os dedos e a mão nas três dimensões, para uma estimativa mais precisa da orientação. Após estes procedimentos, a aceleração, a velocidade angular e o campo magnético são estabelecidos para o correto funcionamento.

O último método operador executado previamente ao envio dos dados referidos à orientação 3D aplica o conceito de quaternions, neste modelo, utilizando a convenção de eixos NED (North East Down) como padrão de direção.

A utilização de quaternions neste projeto faz uso da implementação de Madgwick [16], em linguagem C. Acrescenta-se somente o uso das medidas coletadas dos sensores para realizar uma estimativa de orientação absoluta. A atualização ocorre a cada iteração de repetição por meio de um temporizador da placa microcontroladora.

O cálculo de rastreamento dos dedos utilizou dados de aceleração de corpo obtidos nos três eixos. A partir da aceleração, pode-se obter velocidade e posição, via integração. Porém, calcular a posição através da integral dupla da aceleração de um corpo pode levar ao que é conhecido como integration drift [14].

## 5 RESULTADOS E DISCUSSÕES

No momento que os dados de orientação (em formato de quaternion) são recebidos pela simulação em RV, eles são transformados e vinculados com recursos da classe Transform da Unity. A conversão retorna os ângulos das falanges mediais dos dedos indicador, médio, anelar e mínimo, e da falange proximal do polegar para a mão virtual de um personagem que representa o usuário em uma cena.

Neste personagem, com o auxílio do pacote Animation Rigging na versão 0.3.4-preview, é aplicada cinemática inversa, tornando

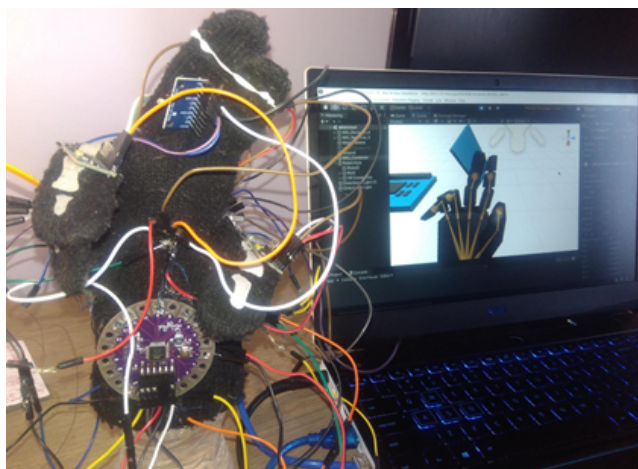


Figura 4: Protótipo VR Wear em fase de simulação.



Figura 5: Aplicação capturando os movimentos da luva, e representando-os em uma mão virtual.

possível a ligação entre as juntas dos dedos e dos braços, possibilitando uma visualização mais realista ao usuário e auxiliando na qualidade da interação com o ambiente de RV.

A simulação implementada abrange apenas a visualização de movimentos da mão. A Figura 4 e a Figura 5 apresentam um ciclo de execução, na qual se realiza um gesto com a mão direita. O protótipo criado é acoplado a uma garrafa plástica, possibilitando testes e coleta de dados com as mãos livres, evidenciando o levantamento de novos requisitos para a continuidade do projeto. Na Figura 5 é possível uma visualização do ponto de vista do usuário em RV com os deslocamentos pré-configurados, enquanto que na Figura 4 demonstra-se o protótipo realizando um gesto.

Como representado pela Figura 2, foram acopladas três baterias, de modo que a distribuição energética foi dividida, aplicando-se cerca de 6V e 200mAh no módulo bluetooth, 3.6V e 20mAh nos escravos  $I^2C$ , e 3.7V e 1100mAh no microcontrolador, utilizando-se um Ground comum entre todos para o devido funcionamento das comunicações. Denota-se que baterias do tipo célula, apesar de serem compactas e atenderem às especificações necessárias, têm pouca capacidade energética.

A confecção do protótipo foi importante para definir opções estruturais e de acoplamento de dispositivos. Nesta luva são utilizados jumpers, já que em testes preliminares linhas condutivas causaram interferência no sinal. Entretanto, o uso de jumpers torna complexa e volumosa a confecção e a ligação entre componentes. Com adesivos termoplásticos, foi possível prender devidamente fios e dispositivos, sem interferência nas conexões elétricas. Porém, a aderência pode acabar se deteriorando com o tempo. Linhas de costura, elásticos ou presilhas poderiam ser usados para fixar componentes, permitindo fácil reconfiguração do equipamento.

## 5.1 Limitações do Estudo

Ao longo dos testes realizados, observou-se um atraso de  $\approx 2$ s não previsto na representação virtual da orientação 3D. Isto se deu pelo fato de que o cálculo e envio de dados de cada quaternion (um para cada dedo representado) é sequencial. Cada IMU aplicada na simulação incrementa esse atraso, gerando latência na visualização.

Em relação ao uso de IMUs, notou-se que o período de um ciclo de envio de dados é muito custoso. Nos testes realizados, sem considerar processamentos adicionais realizados pela aplicação de RV, consumiu-se um tempo de ciclo de 92,7ms para 5 IMUs, e 26,6ms para uma IMU. Isso mostra que ainda é preciso otimizar recursos para reduzir o tempo de processamento, e permitir uma interação com tempo de resposta ao usuário aceitável. Para tanto, recomenda-se a utilização de threads pelo script receptor.

Na programação de microcontroladores, é preciso respeitar os limites impostos à memória de programa. Para isso, faz-se uso apenas de variáveis essenciais, evitam-se redundâncias, e ajustam-se configurações de compilação para otimizar o uso dos registradores. Apesar dos cuidados, a complexidade do código gerado ainda exigiu processamento e uso de memória elevados, comprometendo a possibilidade de execução em tempo real da solução neste momento.

Possíveis soluções para falta de memória e latência consideram usar máquina com maior poder computacional, aplicar conceitos de programação paralela em laços de repetição (se mais de um núcleo de processamento estiver disponível), aumentar taxas de transmissão (como frequência de clock e baudrate), ou simplesmente distribuir trabalho com outras placas microcontroladoras.

Neste quesito, a placa Arduino LilyPad, apesar das facilidades fornecidas, acaba por não ser o microcontrolador ideal para este tipo de solução, não atendendo à resposta em tempo real requisitada, por limitações de hardware. Desta forma, o mesmo poderia ser trocado por outro microcontrolador vestível, como uma versão vestível da placa microcontroladora ESP32, que possui dois núcleos (permitindo multithreading), muito mais memória de programa disponível, e tanto Bluetooth quanto Wi-Fi integrados, evidentemente, por um custo proporcional [24].

## 6 CONCLUSÃO

Este trabalho apresentou o vestível **VR Wear** para rastreamento da mão e dos dedos em projetos de RV na Unity. É um protótipo de custo acessível, baixa latência, compatibilidade entre componentes, e suporte ao monitoramento fisiológico. Ainda é preciso avaliar a usabilidade e o desempenho do equipamento em experimento com usuários. Além disso, indicam-se algumas melhorias: aprimorar as conexões de comunicação e o uso de linhas condutivas de alta

qualidade; incrementar memória para reduzir latência e melhorar o tempo de resposta ao usuário; utilizar baterias LiPo; utilizar IMUs nas ligações dos braços para estimar a posição no rastreamento 3D (conforme Abella e Demircan [1]); adicionar sensação de toque.

## REFERÊNCIAS

- [1] John Abella and Emel Demircan. 2019. A multi-body simulation framework for live motion tracking and analysis within the unity environment. In *2019 16th International Conference on Ubiquitous Robots (UR)*. IEEE, Jeju, Korea, 654–659.
- [2] Apple. 2019. Watch Series 4. <https://www.apple.com/apple-watch-series-4/>
- [3] Arduino. 2018. Datasheet: LilyPad. <https://www.mouser.com/catalog/specsheets/LilyPad.pdf>
- [4] Ke-Yu Chen, Shwetak N. Patel, and Sean Kelle. 2016. Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose, USA, 1504–1514.
- [5] Inrak Choi, Eyal Ofek, Hrvoje Benko, Mike Sinclair, and Christian Holz. 2018. Claw: A multifunctional handheld haptic controller for grasping, touching, and triggering in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montreal, Canada, 1–13.
- [6] James Connolly, Joan Condell, Brendan O'Flynn, Javier Torres Sanchez, and Philip Gardiner. 2017. IMU sensor-based electronic goniometric glove for clinical finger movement analysis. *IEEE Sensors Journal* 18, 3 (2017), 1273–1281.
- [7] Xavier de Tinguy, Thomas Howard, Claudio Pacchierotti, Maud Marchal, and Anatole Lécuyer. 2020. Weatavix: wearable actuated tangibles for virtual reality experiences. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*. Springer, Leiden, Netherlands, 262–270.
- [8] HTC. 2019. VIVE Cosmos. <https://www.vive.com/eu/cosmos>
- [9] InvenSense. 2013. Datasheet: MPU-6000 and MPU-6050 Product Specification Revision 3.4. <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- [10] InvenSense. 2014. Datasheet: MPU9250 Product Specification Revision 1.1. <http://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>
- [11] Itead. 2010. Datasheet: HC-05. <http://www.electronicastudio.com/docs/istd016A.pdf>
- [12] Cédric Kervegant, Félix Raymond, Delphine Graeff, and Julien Castet. 2017. Touch hologram in mid-air. In *ACM SIGGRAPH 2017 Emerging Technologies*. ACM, Los Angeles, USA, 1–2.
- [13] Mingyu Kim, Changyu Jeon, and Jinmo Kim. 2017. A study on immersion and presence of a portable hand haptic system for immersive virtual reality. *Sensors* 17, 5 (2017), 1141.
- [14] Win Tun Latt, Kalyana Chakravarthy Veluvolu, and Wei Tech Ang. 2011. Drift-Free Position Estimation of Periodic or Quasi-Periodic Motion Using Inertial Sensors. *Sensors* 11, 6 (2011), 5931–5951. <https://doi.org/10.3390/s110605931>
- [15] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. 2017. *3D user interfaces: theory and practice*. Addison-Wesley Professional, New York, USA.
- [16] Sebastian Madgwick. 2012. Open source IMU and AHRS algorithms. <https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/>
- [17] Steve Mann. 1998. Humanistic computing: 'WearComp' as a new framework and application for intelligent signal processing. *Proc. IEEE* 86, 11 (1998), 2123–2151.
- [18] Maxim. 2014. Datasheet: MAX30100. <https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf>
- [19] Luca Micelli, David Acosta, Alvaro Uribe-Quevedo, Fabrizio Lamberti, and Bill Kapralos. 2018. Extending Upper Limb User Interactions in AR, VR and MR Headsets Employing a Custom-Made Wearable Device. In *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, Zakynthos, Greece, 1–4.
- [20] Claudio Pacchierotti, Stephen Sinclair, Massimiliano Solazzi, Antonio Frisoli, Vincent Hayward, and Domenico Prattichizzo. 2017. Wearable haptic systems for the fingertip and the hand: taxonomy, review, and perspectives. *IEEE transactions on haptics* 10, 4 (2017), 580–600.
- [21] Robert J Seidel and Paul R Chatelier. 2013. *Virtual Reality, Training's Future?: Perspectives on Virtual Reality and Related Emerging Technologies*. Springer, New York, USA.
- [22] Simon Tatham. 1999. PuTTY. <https://www.putty.org/>
- [23] Unity. 2021. Unity Real-Time Development Platform | 3D, 2D VR and AR Engine. <https://unity.com>
- [24] Velleman. 2020. Esp32 wearable development board. <https://www.velleman.eu/products/view/?id=449192>
- [25] Mariusz P Wilk, Javier Torres-Sanchez, Salvatore Tedesco, and Brendan O'Flynn. 2018. Wearable human computer interface for control within immersive VAMR gaming environments using data glove and hand gestures. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*. IEEE, Galway, Ireland, 1–9.