

Relatório Técnico

Tesseract

Angelo Borsoi Ross – angeloborsoiross@hotmail.com
Davi Boberg – daviboberg@alunos.utfpr.edu.br
Gustavo Timmermans Pires da Silva – gustimmer@gmail.com
Lucas Kaminski de Freitas – lucaskfreitas@hotmail.com

Novembro de 2018

Resumo

O projeto Tesseract é um sistema versátil de reprodução de áudio controlado remotamente por um dispositivo Android. Seu principal diferencial é um sistema físico de visualização de áudio construído utilizando LEDs e *infinity mirrors*. Para isso foi utilizado uma impressão 3D, LEDs programáveis e um Raspberry Pi 3B, com intuito de permitir a customização do projeto e integração à plataforma de músicas Spotify, permitindo a seus utilizadores escutarem e compartilharem com os demais suas músicas favoritas de uma maneira que vai além da simples audição.

1 Introdução

Tesseract é um sistema versátil de reprodução de áudio desenvolvido para proporcionar ao usuário uma experiência agradável através de sua interface inovadora, de fácil uso e pela sua estética.

O sistema como um todo é formado por duas partes: um aplicativo Android e o Tesseract em si, que é uma caixa de som em formato de cubo. Uma visão geral do sistema está representada no diagrama da Figura 1.

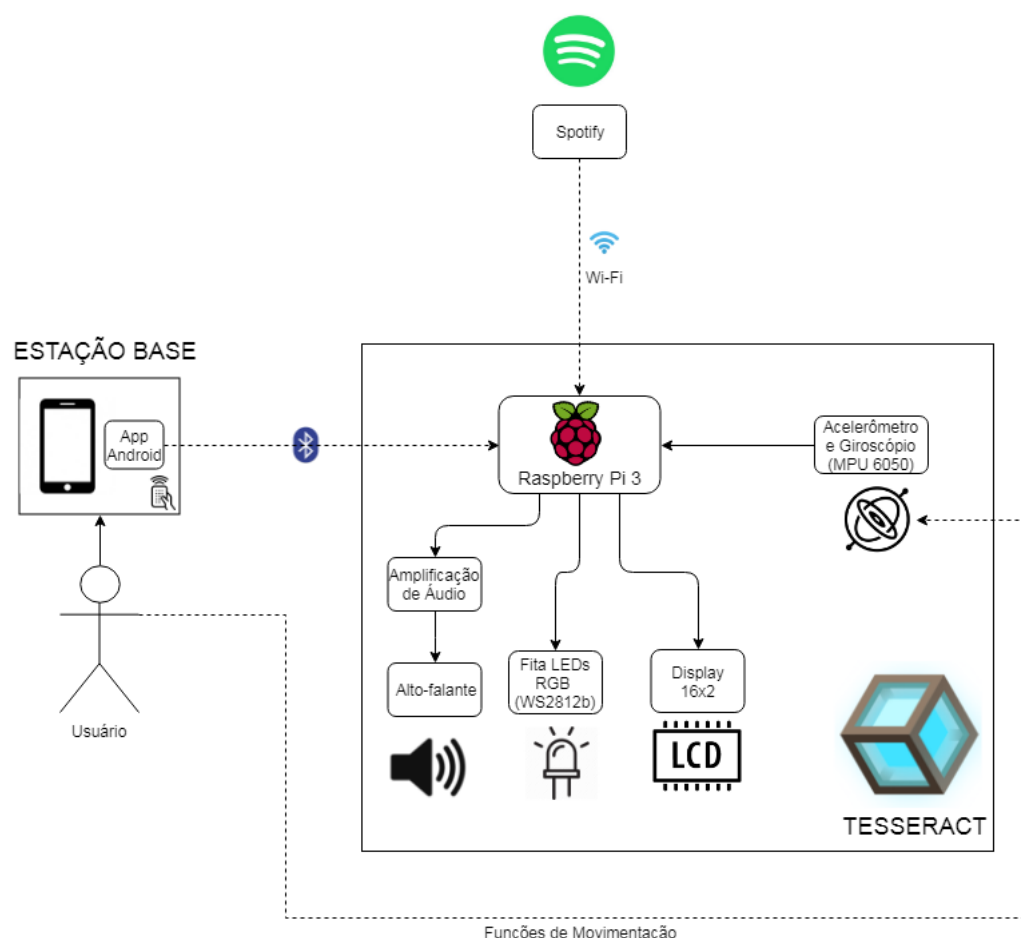


Figura 1: Diagrama representando de forma geral o projeto Tesseract

Após instalar o aplicativo em seu dispositivo Android, o usuário pode transmitir e tocar no *Tesseract* todas as suas listas de músicas. Caso deseje, também é possível realizar a conexão do *Tesseract* diretamente à plataforma de músicas Spotify através da internet (Wi-Fi). Com isso, passa a ser possível deixar várias músicas tocando sem a necessidade do intermédio do celular. Essa funcionalidade é desenvolvida apenas para a plataforma Spotify.

Caso o usuário preferir, também é possível navegar na lista de reprodução (seja do celular ou do Spotify) interagindo diretamente com o *Tesseract*. Isso é possível graças a um sensor de movimentos presente no sistema. Através de movimentos específicos, como levantar e abaixar, é possível executar as funções de próxima/anterior, parar/tocar e embaralhar a lista de reprodução.

O *Tesseract* também conta com *infinity mirrors* nas faces laterais, como o da figura 2, feita com LEDs RGB. Através do controle desses LEDs, é feita uma "visualização da música", ou seja, os LEDs acendem, apagam e trocam de cor reagindo à intensidade e frequência (agudos e graves) da música. O comporta-

mento exato dos LEDs de cada face pode ser configurado pelo usuário também através do aplicativo.



Figura 2: Infinity Mirror
Fonte: [Inovation & Decorations 2016]

Mais informações, detalhes, imagens e um vídeo demonstrativo do funcionamento do *Tesseract* podem ser encontrados no blog mantido pela equipe ao longo do desenvolvimento do projeto:

<https://projetotesseract.wordpress.com>

Na seção 2 são apresentadas de maneira breve as principais tecnologias utilizadas. Na seção 3 é mostrado como cada uma das tecnologias é utilizada para viabilizar o projeto.

2 Tecnologias utilizadas

Nesta seção são descritas brevemente algumas das tecnologias utilizadas no projeto. O objetivo dessa seção é apresentar de maneira sucinta as características principais das tecnologias mais importantes utilizadas.

2.1 Raspberry Pi 3 Model B

Segundo a [Raspberry Pi Foundation 2016], o *Raspberry Pi 3 Model B*, representado na Figura 3 é o primeiro modelo da terceira geração do *Raspberry Pi*. Ele substituiu o *Raspberry Pi 2 Model B* em fevereiro de 2016.

Quanto às especificações, possui um processador de quatro núcleos, 64 bits de 1.2GHz: um Broadcom BCM2837. Também possui 1GB de RAM, uma placa de rede BCM43438, que suporta redes *Wi-Fi* e *Bluetooth* de baixa energia.



Figura 3: Um *Raspberry Pi 3 Model B*
Fonte: [Raspberry Pi Foundation 2016]

Para periféricos, possui 4 portas USB 2, saída de áudio, saída HDMI para áudio e vídeo, conector para cartões Micro SD e conectores para conectar uma câmera e uma tela *touchscreen*. Além disso, possui 40 pinos de entrada e saída genéricos para conexões com demais dispositivos.

2.2 Acelerômetro/Giroscópio MPU-6050

O *MPU-6050* é um dispositivo projetado para usos de baixa energia, baixo custo e alta performance como em *smartphones*, *tablets* e sensores vestíveis. Ele combina um giroscópio de 3 eixos e um acelerômetro de 3 eixos juntamente com um processador de movimento digital [InvenSense 2018].

Seu giroscópio tem capacidade de medição de até $2000^\circ/s$ e seu acelerômetro tem escala de até 16 g. Para comunicação, utiliza *I²C* e sua tensão de operação fica entre 2,375 V e 3,46 V. A Figura 4 mostra uma pequena PCB que faz uso do *MPU-6050*.

2.3 Fita de LEDs WS2812B

A fita de LEDs nada mais é do que vários LEDs *WS2812B* conectados em cascata. Tais LEDs possuem um pixel de cada cor primária, vermelho, verde e azul, os quais podem representar 256 intensidades diferentes cada. Cada LED possui um circuito integrado responsável por seu controle. A comunicação é feita seriamente por um único barramento de controle, em que cada circuito integrado obtém um comando e propaga os demais para os circuitos seguintes da fita.

O circuito opera com uma tensão entre 3,5 V e 5,3 V e comandos podem ser enviados com uma taxa de transmissão de até 800 *kbps* [Worldsemi 2016]. A Figura 5 mostra como é a fita composta pelos LEDs *WS2812B*.

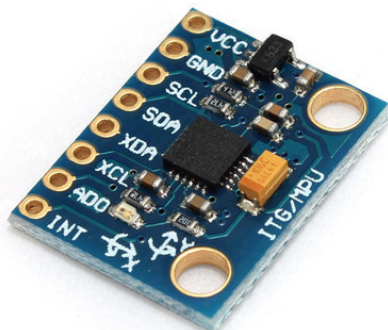


Figura 4: PCB utilizando um MPU-6050

Fonte: [FilipeFlop 2018]



Figura 5: Fita de LEDs WS2812B

2.4 Bluetooth Low Energy

O *Bluetooth* é uma tecnologia de transmissão de dados sem fio. A versão “*Low Energy*” é uma especificação modificada com o objetivo de diminuir o consumo energético, mas mantendo sua capacidade de transmissão.

O *Bluetooth Low Energy* tem um alcance teórico máximo de 150 metros em campo aberto, mas suas aplicações práticas geralmente o limitam a poucos metros. Utiliza frequências ao redor de 2,4 GHz para comunicação e seus módulos devem ter um uso máximo de corrente de 15 mA e latência de até 3 ms. Quando ocioso, o módulo deve ter um uso de corrente de aproximadamente 1 μ A, tornando-o ideal para aplicações que dependam do uso de bateria ou tenham limitação na quantidade de corrente [Decuir 2014].

2.5 Wi-Fi

Wi-Fi se refere aos meios de comunicação sem fio baseados nos padrões IEEE 802.11. O sistema *Wi-Fi* que será utilizado no projeto segue o a especificação 802.11N. Tal padrão foi aprovado em 2009, sendo uma melhoria sobre os anteriormente aprovados (802.11A, 802.11B e 802.11G).

Permite uma taxa de transmissão de dados máxima de aproximadamente 600 *Mbps*, opera nas frequências de 2,4 ou 5 GHz e possui uma largura de canal nominal de 20 ou 40 MHz [Poole 2018].

2.6 PAM8403

O *PAM8403* é um amplificador de áudio estéreo de 3 *W*, classe D. Possui eficiência energética de até 90%, proteção contra curto circuito, desligamento automático ao sobreaquecer e saída de ruído baixa. Opera com uma tensão entre 2,5 *V* e 5,5 *V* e possui uma demanda de corrente de 16 *mA* quando ocioso. A Figura 6 mostra um módulo que utiliza o *PAM8403*.

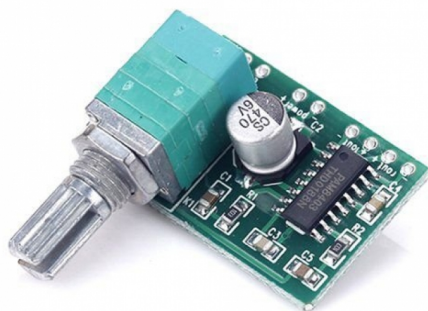


Figura 6: Módulo que utiliza o *PAM8403*
Fonte: [Amazon 2018]

2.7 Android

O Android é um sistema operacional da Google® destinado a dispositivos celulares. Atualmente em sua versão 9.0, o sistema baseado em Linux conta com mais de 3 milhões de aplicativos em sua loja virtual [statista 2018].

Em 2017 se tornou o sistema operacional mais utilizado para acesso a internet do mundo, de acordo com a pesquisa realizada pela StatCounter [StatCounter 2017].

2.8 Transformada rápida de Fourier

Transformada rápida de Fourier (em inglês *Fast Fourier Transform*, ou FFT) é um algoritmo eficiente para se calcular a transformada discreta de Fourier (DFT) e a sua inversa. A DFT converte uma sequência finita de amostras igualmente espaçadas de uma função em outra sequência de amostras complexas igualmente espaçadas que representam suas componentes em frequência [Loan 1992].

2.9 LCD (HD44780)

O LCD 16x2 é controlado pelo microcontrolador HD44780. É capaz de reproduzir dígitos alfanuméricos, caracteres japoneses *kana* e símbolos. Entre suas principais características, o HD44780 suporta tensão de operação ente 2,7 V a 5,5 V, pode reproduzir cada símbolo em matrizes de pontos de 5x8 ou 5x10, suporta operação em 4 bits ou 8 bits. Além disso é capaz de armazenar até 80 caracteres em sua memória RAM [Hitachi 2018].

3 Desenvolvimento

O desenvolvimento do projeto é separado em cinco partes:

- Estrutura física;
- Hardware, que comporta o software e/ou é acoplado à estrutura;
- Aplicativo Android;
- Raspberry Pi;
- Comunicação sem fio.

Cada uma dessas partes será agora descrita separadamente.

3.1 Estrutura física

A estrutura física foi primeiramente desenvolvida em um modelo 3D utilizando o software SketchUp¹ e então impressa em uma impressora 3D, a qual tinha um espaço de operação de até 20x20x20cm. A estrutura final possui um tamanho total de 16x16x16cm e foi feita com um filamento PetG translúcido, sendo separada em uma base e uma tampa.

Na base da estrutura, quatro pilares verticais com 3cm de comprimento e largura e 13cm de altura são conectados horizontalmente na parte inferior, formando uma espécie de chassi. Pequenas concavidades permitem assentar dois painéis de acrílico, de 10x10x0.3cm, por face, os quais ficam firmemente acomodados devido ao atrito com o plástico e paralelos em relação um ao outro.

¹<https://www.sketchup.com/>

Nesses painéis de acrílico, instalou-se insulfilme, o que gera um efeito similar ao espelho quando há forte luz em um dos lados do painel. Usando um pequeno jogo de luzes, criou-se um efeito “espelho infinito”². A partir desse momento, o conjunto de espelho infinito com LEDs (os quais serão tratados adiante) será chamado de “*infinity mirror*”. O interior da base também abriga todos os periféricos com exceção do alto-falante e LCD, os quais ficam presos à tampa.

A tampa da estrutura possui aberturas feitas para afixar o alto-falante e o LCD, além de pequenos pinos de fixação e buracos para parafusos, os quais são usados para prender não só os periféricos mas também a tampa à base. Pequenas concavidades na parte inferior auxiliam na acomodação dos painéis de acrílico.

Vale notar que a impressora 3D utilizada possui precisão de aproximadamente 0.5mm, por esse motivo várias modificações foram feitas ao modelo 3D inicial a fim de prepará-lo para impressão, tais como alargamento de concavidades e mudanças nos sistemas de conexão base-tampa. Apesar dessa dificuldade, a estrutura impressa 3D atende aos requisitos do projeto. A Figura 7 mostra a estrutura com os acrílicos montados, sem o efeitos dos LEDs.



Figura 7: Estrutura montada do Tesseract

²https://pt.wikipedia.org/wiki/Espelho_infinito

3.2 Hardware

A fita de LEDs *WS2812* foi cortada em diversos segmentos de 5 LEDs. Em cada uma das faces laterais do Tesseract, foram colados quatro desses segmentos, um em cada aresta, totalizando 80 LEDs no projeto inteiro. A alimentação de cada face foi realizada de forma separada, mas o barramento de controle é apenas um para todas as faces. Ou seja, a saída de uma face foi conectada diretamente à entrada da face seguinte, e assim por diante. Dessa forma, do ponto de vista do Raspberry Pi, existe apenas uma fita.

A figura 8 mostra as faces do Tesseract sem as placas de acrílico, mas com os LEDs colados às arestas.

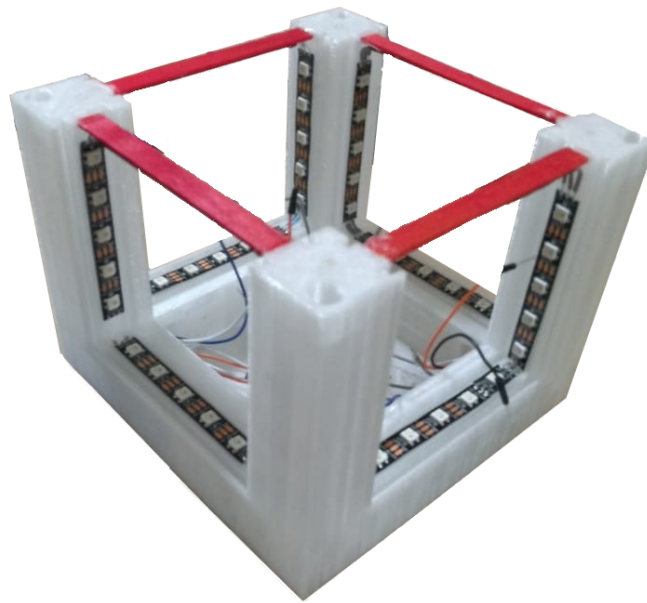


Figura 8: LEDs afixados à estrutura do Tesseract

A conexão com o módulo acelerômetro *MPU6050* com o Raspberry Pi 3 foi feita diretamente através de sua interface *I²C* nativa. Apesar do módulo apresentar um pino para gerar interrupções, foi decidido não utilizá-lo, pois isso poderia comprometer o tempo de resposta das outras tarefas mais prioritárias no Raspberry.

O circuito de áudio, localizado em uma PCB, consiste em conectar o *Headphone Jack* do Raspberry Pi ao alto-falante de 3 W que fica preso à tampa do Tesseract através de um amplificador *PAM8403*. Como a saída de áudio do Raspberry é estéreo, o lado direito e o lado esquerdo precisam ser somados para que o alto-falante possa reproduzir todos os sons. Para isso, foi necessário juntar os dois sinais na entrada do módulo amplificador, utilizando-o como um circuito

somador simples³.

O LCD foi conectado ao Raspberry em modo 4 pinos. Para regular seu contraste, foi utilizado um divisor de tensão simples, localizado na PCB junto ao circuito de áudio.

Por fim, a PCB também é responsável pela distribuição da energia. Através de uma barra de pinos simples, a fonte de alimentação é conectada diretamente à PCB, que apresenta outras barras de pinos. Através dessas, todos os dispositivos dentro do Tesseract podem ser alimentados.

As figuras 9 e 10 mostram o esquemático do circuito elétrico e a PCB confeccionada, respectivamente.

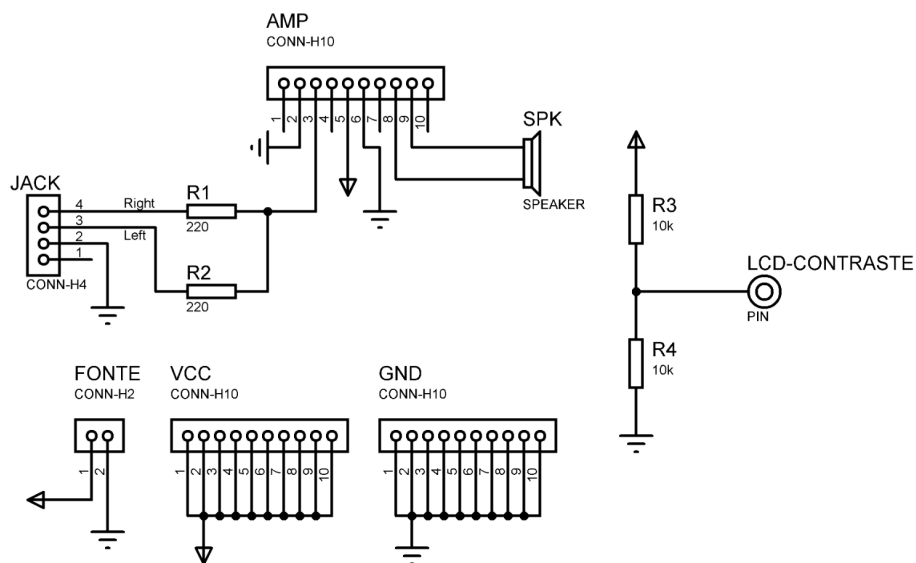


Figura 9: Esquemático da PCB, que consiste em barras de pinos para alimentação, encaixe do módulo amplificador e o pino para o contraste do LCD

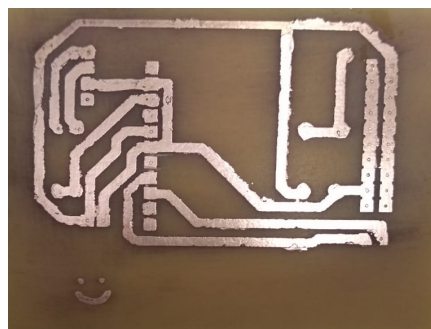


Figura 10: PCB confeccionada

³https://www.electronics-tutorials.ws/opamp/opamp_4.html

3.3 Aplicativo Android

O aplicativo Android, estação base do projeto, tem a responsabilidade de controlar e configurar diversas funcionalidades do Tesseract, entre elas o controle da música, a conexão do Raspberry ao Wifi, seleção e edição de padrões exibidos nos *infinity mirrors*, realizar conexão ao Spotify e seleção da lista de músicas a ser reproduzida. A figura 11 mostra a tela principal do aplicativo Android.

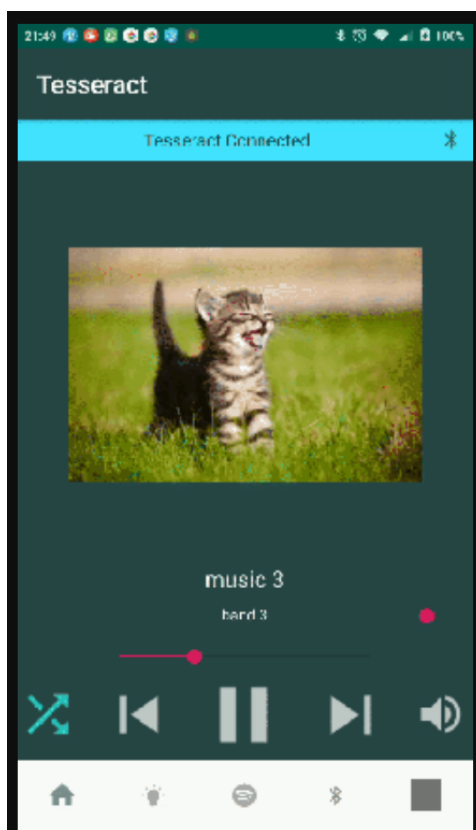


Figura 11: Tela principal do aplicativo Android

A aplicação foi desenvolvida utilizando-se a arquitetura *model-view-controller* (MVC).

Abaixo estão escritos em mais detalhes como são realizados o controles de algumas das funções.

3.3.1 Conexão e seleção de lista de músicas do Spotify

Uma opção no aplicativo permite ao usuário se conectar à sua conta do Spotify. A autenticação do usuário é feita através da API do serviço. Uma vez conectado, passa a ser possível tocar músicas do Spotify no Tesseract. A seleção e a navegação da lista de músicas pode ser realizada totalmente através do aplicativo.

Simultaneamente, ao realizar a conexão através do aplicativo, também é realizada a conexão do próprio Tesseract diretamente ao servidor do Spotify. Dessa forma, mesmo que a conexão entre a estação base e o Raspberry Pi seja cortada, a música continuará sendo tocada através do alto falante do Tesseract.

3.3.2 Seleção e edição de padrões

Os padrões básicos são armazenados no Tesseract e enviados através do Bluetooth ao aplicativo quando requisitados. No aplicativo o usuário pode escolher entre os diferentes padrões existentes, escolhendo também a face do cubo na qual será utilizado. A personalização pode ser feita a partir do padrões básicos, definindo as cores das diferentes componentes disponíveis para o padrão base selecionado. Novos padrões criados são armazenados no dispositivo celular que executa o aplicativo.

3.3.3 Controle da música

O aplicativo permite ao usuário controlar as músicas sendo reproduzidas, sendo possível avançar para a próxima, retorna para a anterior, pausar ou iniciar, alterar o volume e selecionar a opção de ordem aleatória (*shuffle*). Além disso, informações sobre a música sendo reproduzida no Tesseract são enviadas ao aplicativo, de modo que o usuário da estação base saiba o que está sendo tocado.

3.4 Raspberry Pi

As funcionalidades do Tesseract foram todas desenvolvidas utilizando um Raspberry Pi 3 modelo B. Abaixo estão descritas as diversas tarefas executadas por ele.

3.4.1 Saída de áudio

Como já mencionado anteriormente, a saída de áudio utilizada pelo Tesseract é a própria saída do Raspberry. Para o desenvolvimento do projeto, foi necessário instalar o servidor de áudio *PulseAudio*. Através dele, foi criada uma interface *loopback* que retorna todo o áudio que for tocado pelo Raspberry. Essa interface é necessária para que as informações do áudio pudessem ser utilizadas pelos algoritmos que serão descritos a seguir.

3.4.2 Análise de frequência

De modo a permitir que os LEDs presentes na faces com *infinity mirrors* reajam seguindo algumas características da música sendo reproduzida, é utilizado o algoritmo de transformada rápida de Fourier (FFT) para obtermos as componen-

tes de frequência do áudio. Para obter uma melhor resposta do sinal processado é utilizado o janelamento Hanning⁴.

3.4.3 LCD

O funcionamento do LCD é simples. Foi utilizado a biblioteca *RPLCD* para Python juntamente com o controle dos pinos de propósito geral. O dispositivo mostra em sua tela informações relevantes ao utilizador, como uma mensagem aguardando a conexão por Bluetooth ou o nome da música e artista ou banda que estão sendo tocados no momento.

3.4.4 Detecção de movimentos

A detecção de movimentos é feita com o acelerômetro. Como esse sensor retorna valores de aceleração instantânea, os movimentos desejados (inclinado para esquerda ou direita, levantar e abaixar e sacudir) devem ser descritos dessa maneira.

Os movimentos de inclinação foram descritos como uma aceleração no eixo X e eixo Y, resultando numa rotação, uma parada e então um retorno a posição normal. O movimento de levantar e abaixar foi descrito como uma aceleração apenas no eixo Y, uma parada e então uma nova aceleração apenas no eixo Y. Finalmente, ação de sacudir foi detectada calculando a magnitude do vetor aceleração e comparando-o com um limite mínimo. Se o limite mínimo é alcançado, um contador é incrementado e, após um pequeno número de incrementos, a ação considerada uma “sacudida”.

Deve-se notar que o acelerômetro foi posicionado de lado dentro do cubo, sendo assim o eixo X e Y reais foram trocados no código.

3.4.5 Controle da fita de LEDs

Uma característica importante do funcionamento da fita é que para atualizar apenas um LED, devem ser enviados informações para todos os outros LEDs que estão à frente do mesmo.

Para realizar a atualização do estado de um LED na fita, é necessário, em média, 30us. Como o projeto envolve o controle de 80 LEDs e, no pior caso, devemos atualizar todos os 80 LEDs, cada atualização pode levar 2.4ms.

Tendo em vista a limitação acima, o desenvolvimento da biblioteca de controle das funções de iluminação se preocupou em fazer os cálculos durante a etapa de atualização o mais rápido possível, usando muitas vezes valores pré-computados. Para fazer um sistema versátil que pudesse ser expandido com novas funções facilmente, o processo de criação de uma função foi dividida em três partes:

⁴Mais pode ser lido sobre FFT e janelas em <http://www.ni.com/white-paper/4844/pt/>

- **Geração de cores:** A sequência de cores a ser usada pela função. O sistema oferece duas funções de geração de gradiente.
- **Preparação:** Essa etapa é particular a cada função. São pré-computados aqui quaisquer valores necessários para a atualização da função.
- **Execução:** A cada intervalo de tempo configurável t , o sistema computa uma atualização da função e a deixa preparada para ser transmitida à fita.

Foram desenvolvidas diversas configurações de visualização, entre elas:

- *Wave*: Se assemelha a uma onda passando pela fita, com diversas velocidades e número de ondas.
- *Stream*: Tenta imitar um córrego com as cores sendo atualizadas.
- *Breathe*: A intensidade da cor de cada um dos LEDs é modificada por uma função senoidal configurável.
- *Johnson*: Similar a um contador *Johnson*. Possui três modos que alteram seu comportamento.
- *ColorShift*: Uma sequência de cores numa intensidade constante.
- *FFT1*: É usado o valor retornado da *FFT* para modificar o tom de cor de um LED de acordo com uma intensidade harmônica. Pode ser dividido em n segmentos do espectro de frequência, onde n só é limitado pelo número de LEDs numa face.
- *FFTbar*: É usado o valor retornado da *FFT* para modificar o número de LEDs acesos em cada uma das arestas de uma face.
- *FFTint*: É usado o valor retornado da *FFT* para modificar a intensidade de uma sequência de cores qualquer.

3.5 Comunicação

Dois sistemas de comunicação foram utilizados no Tesseract, ambos responsabilidade do Raspberry Pi utilizando suas interfaces nativas. A primeira é uma comunicação Bluetooth entre o aplicativo e o Raspberry. A segunda é entre o Raspberry e o servidor da plataforma Spotify por meio da internet.

3.5.1 Bluetooth

A comunicação entre o Raspberry e a estação base é realizada através da arquitetura de rede cliente-servidor, sendo o Tesseract atuando como servidor. Enquanto nenhum dispositivo está conectado ao Tesseract, o Bluetooth permanece detectável e aguardando uma conexão. Após ser estabelecida uma comunicação, o Tesseract fica escutando e enviando mensagens à estação base até que a comunicação seja encerrada.

3.5.2 Plataforma Spotify

Diversas APIs diferentes da plataforma Spotify foram necessárias durante o desenvolvimento do projeto:

- Android SDK: Instalada no aplicativo Android, essa API fornece ferramentas de autenticação do usuário e do aplicativo em questão.
- Web API: Após uma certa configuração, a Web API permite o controle e a navegação de uma lista de músicas através de requisições HTTP enviadas diretamente para o servidor do Spotify. Essa API foi utilizada tanto pelo aplicativo quanto pelo Raspberry Pi.
- Spotify Connect: Foi utilizada a biblioteca *raspotify*⁵ para que o Raspberry passasse a funcionar como um cliente da plataforma. É através dessa biblioteca que a música é efetivamente tocada pelo Tesseract.

Além das APIs, também foi utilizado o módulo *wifi* para Python, que permitiu a manipulação da interface Wi-Fi do Raspberry Pi através do código, permitindo então uma conexão estável e de banda larga com o Spotify. Deve-se notar que o módulo não funcionou adequadamente por maneira própria, então foi feita uma modificação à um conserto encontrado em um fórum popular na internet. Tal modificação e o devido crédito estão disponíveis no apêndice.

4 Conclusão

O projeto Tesseract apresentou durante seu desenvolvimento o desafio de integrar vários módulos de ambos *hardware* e *software*, abrangendo desde periféricos simples até dispositivos embarcados potentes com seus próprios sistemas operacionais. Tal integração permitiu confeccionar um produto final que inclui as mais diversas áreas dentro de um curso de engenharia da computação, como sistemas embarcados, redes de computadores e várias instâncias de desenvolvimento de software.

Aprendeu-se muito sobre o desenvolvimento de um grande projeto, porém alguns dos conhecimentos mais valiosos que foram adquiridos não estão relacionados à área técnica, e sim ao planejamento, gerência e documentação do trabalho. Com isso, notou-se a importância de um cronograma, planejamento de riscos (o qual, inicialmente, causou a alteração de vários componentes do Tesseract com o intuito de evitar problemas) e boa comunicação e integração da equipe.

Foram enfrentadas várias dificuldades durante o desenvolvimento, incluindo mas não limitado à falta de experiência da equipe com mecânica, bibliotecas incompatíveis, poder de processamento limitado do *hardware* e a falta de tempo

⁵<https://github.com/dtcooper/raspotify>

para melhor aperfeiçoar o projeto. Em geral, acredita-se que a superação de tais adversidades contribuiu para o crescimento dos membros da equipe e proveu valioso conhecimento para projetos futuros.

Referências

- [Decuir 2014]DECUIR, J. *Bluetooth 4.0: Low Energy*. 2014. Disponível em: <<https://californiaconsultants.org/wp-content/uploads/2014/05/CNSV-1205-Decuir.pdf>>. Acesso em: 23 de setembro de 2018.
- [Hitachi 2018]HITACHI. *HD44780*. 2018. Disponível em: <<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>>. Acesso em: 27 de setembro de 2018.
- [Inovation & Decorations 2016]INOVATION & DECORATIONS. *Infinity Mirror*. 2016. Disponível em: <<https://inovodecor.com/infinity-light-mirror-for-house.html>>. Acesso em: 23 de setembro de 2018.
- [InvenSense 2018]INVENSENSE. *MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking™ Devices*. 2018. Disponível em: <<https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>>. Acesso em: 23 de setembro de 2018.
- [Loan 1992]LOAN, C. V. *Computational Frameworks for the Fast Fourier Transform*. [S.l.]: Cornell University, Ithaca, New York, 1992.
- [Poole 2018]POOLE, I. *IEEE 802.11 Wi-Fi Standards*. 2018. Disponível em: <<https://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11-standards-tutorial.php>>. Acesso em: 23 de setembro de 2018.
- [Raspberry Pi Foundation 2016]RASPBERRY PI FOUNDATION. *Raspberry Pi 3 Model B*. 2016. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Acesso em: 23 de setembro de 2018.
- [StatCounter 2017]STATCOUNTER. *Sistemas Operacionais mais utilizados*. 2017. Disponível em: <<http://gs.statcounter.com/press/android-overtakes-windows-for-first-time>>. Acesso em: 27 de setembro de 2018.
- [statista 2018]STATISTA. *Play Store*. 2018. Disponível em: <<https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>>. Acesso em: 27 de setembro de 2018.
- [Worldsemi 2016]WORLDSEMI. *WS2812B Datasheet and Specifications*. 2016. Disponível em: <<https://www.kitronik.co.uk/pdf/WS2812B-LED-datasheet.pdf>>. Acesso em: 23 de setembro de 2018.

Apêndice

Adequação à WPA Supplicant para o módulo *wifi* para Python

O crédito ao conserto original deve ser dado ao usuário *Captain Fantastic* na seguinte página, no popular serviço *Stack Overflow*: <https://stackoverflow.com/questions/20470626/python-script-for-raspberrypi-to-connect-wifi-automatically>.

A modificação feita pelos autores é grande demais para ser incluída neste documento. Por esse motivo, ela está disponibilizada no repositório online do projeto Tesseract: https://raw.githubusercontent.com/Angelin01/Oficinas-3/master/Tesseract/Communication/scheme_wpa.py