

Police VR Robot Car With Articulated Cameras And Live Stream Video

Arthur Faria Campos*, 16/0024242, Abhay Caran Das Baraky Dias†, 16/0022479

*† Engenharia Eletrônica, UNB-FGA, Brasília, Brasil

Resumo—Este artigo tem como objetivo descrever uma ferramenta policial eletrônica, Police VR Robot Car, para ajudar com o problema envolvendo a violência terrorista, organizada e desorganizada. O projeto integra um carro robô controlado com câmeras estereoscópicas articuladas e transmite seu vídeo para um óculos de realidade virtual conectado a um smartphone. Ambos, a transmissão de vídeo e o controle do carro do robô, são configurados com um servidor gerado pelo Raspberry Pi 3 B+. Procurando atuar em casos como o da mesquita da Nova Zelândia, em 19/03/2019, que permitiria localizar o criminoso com grande eficiência.

Index Terms—Virtual Reality, Stereoscopic Cameras, Raspberry Pi, Micro-controllers, Electronic Police Tool.

I. INTRODUÇÃO

A Indústria de robótica representa um importante mercado no mundo, capaz de gerar, além do dinheiro, grandes inovações uma vez que reúne diversas tecnologias com o objetivo de ajudar a sociedade. Neste sentido, é importante notar o espaço que a realidade virtual (VR) tomou no cotidiano das pessoas nos últimos anos e a revolução que ocorre diariamente com as suas aplicabilidades. Em especial, os sistemas de monitoramento por vídeo avançaram para um novo nível e é neste âmbito que este trabalho busca mais uma nova aplicação para a realidade virtual. [1]

A. Contexto

Esta segunda década do século XXI, que caminha para o seu término, marcou a história pelo refortecimento do movimento terrorista no mundo. Esta premissa é clara ao analisarmos o mapa da tendência terrorista global que possui 3 grandes picos de mortes ocasionadas por terrorismo. O primeiro e o segundo decorreram ao ataque às torres gêmeas em 11 de setembro de 2001, nos Estados Unidos, e ao extenso conflito com o Iraque nos anos seguintes. O terceiro pico ocorreu em 2014 e corresponde às consequências da primavera árabe no Iraque, o surgimento do Estado Islâmico e à guerra da Síria que juntos corroboraram com um aumento de 350 por cento nas mortes por terrorismo, segundo o Global Terrorism Index [4].

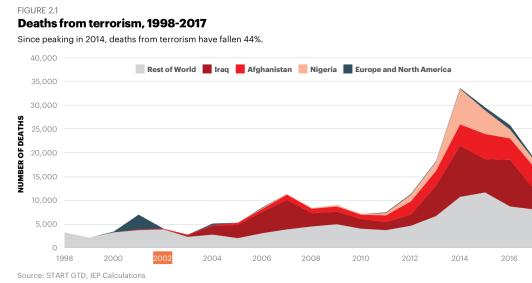


Figura 1. Mapa das mortes por terrorismo no mundo.

Paralelamente, os recentes anos se caracterizaram pelo aumento esporádico de atos terroristas motivados por xenofobia e islamofobia, fato que pode ser relacionado às grandes movimentações de refugiados pelo mundo por conta da guerra na Síria e pelas atividades terroristas de grupos como o Boko Haram. No dia 15 de março deste ano de 2019, um extremista realizou um ataque a uma mesquita na Nova Zelândia e transmitiu toda a sua atuação por uma live stream em sua rede social. O ato foi planejado por mais de dois anos e o autor chegou a escrever um manifesto anti-islã para justificar a sua atitude.

B. Realidade Virtual

Por consequência, o mundo vive hoje uma corrida para criar tecnologias que ofereçam segurança e inovação no combate ao terrorismo. Um ótimo exemplo são os drones policiais que possuem como função a coleta de dados, mapeamento de locais e, por vezes, auxílio na neutralização de criminosos. Neste sentido, o designer Jin-Jung Young criou o Drone da Polícia da Audi Patrone. A principal característica do drone é o sistema VR e o headset, que permitem que a polícia o controle a uma distância segura. Suas câmeras de alta resolução dão à polícia uma visão clara da cidade. O drone também possui áudio sensível, luzes poderosas e um sistema de propulsão a jato.

A realidade aumentada e virtual cresceu 17% como uma indústria em 2017, e a taxa de crescimento continuará a aumentar nos próximos cinco anos, conforme os usos práticos da tecnologia continuem a ser desenvolvidos e os usos existentes alcancem maior maturidade. E como parte desta tendência, este trabalho idealiza uma ferramenta policial que unifica a realidade virtual, com câmeras estereoscópicas e um carro robô controlado remotamente.

C. Benefícios

É como no atentado na mesquita da Nova Zelândia que o Police VR Robot Car procura atuar. O robô permitiria localizar o criminoso dentro da mesquita com grande eficiência, uma vez que este utiliza realidade virtual para dar ao policial usuário maior referência espacial e trazer maior segurança para equipe.

II. OBJETIVOS

O Police VR Robot Car tem como objetivo criar um instrumento sólido para localização de terroristas em construções fechadas de modo que ofereça mais segurança para os agentes da polícia neutralizarem o indivíduo durante o seu ato de terror.

A. Qualidade de Vídeo

A qualidade de vídeo entregue pelo sistema será de baixa para média, uma vez que seu objetivo é apenas a identificação dos terroristas em um novo ambiente.

B. Transmissão

O Police VR Robot contará com uma alta taxa de atualização para a live stream tanto de vídeo quanto para o controle das articulações das câmeras, de forma que a transmissão minimize ao máximo o delay.

C. Qualidade de Sinal

Para o produto final deste trabalho, o produto não contará com uma boa qualidade de sinal quando utilizado para grandes distâncias.

D. Estrutura

O projeto terá uma estrutura sólida e robusta e que seja ideal para terrenos planos.

III. METODOLOGIA

Para facilitar o desenvolvimento do protótipo o projeto será dividido em três áreas de trabalho: Transmissão, controle e estrutura. Sendo que, na etapa final do projeto realizaremos testes de viabilidade.

Também contará com repositórios através da plataforma GitHub a fim de facilitar a organização e armazenagem dos produtos e documentos do projeto.

A. Transmissão

A área de Transmissão será o foco principal do projeto, contará com um system on a chip (SoC), Raspberry pi 3 +, para realizar toda a comunicação entre o óculos de realidade virtual e as câmeras e fixadas no veículo.

B. Controle

Está área ficará responsável pela elaboração do controle do movimento das câmeras baseada nos dados coletados do acelerômetro e do magnetômetro do smartphone e também dos controles dos motores do veículo.

C. Estrutura

O foco da área de estruturas é elaborar toda a parte mecânica do projeto, principalmente onde será alocado os motores, os suportes das câmeras e a Raspberry pi.

IV. REQUISITOS

A. Requisitos técnicos

a) *Formatação dos documentos:* A elaboração e manutenção dos documentos produzidos no projeto deverá utilizar LaTeX de forma que a apresentação das informações fique organizada. Assim como, representará as instruções para a construção do protótipo.

b) *Custo:* O projeto deve ser viável economicamente para o escopo da disciplina e restrições da universidade.

B. Requisitos funcionais

a) *Live Stream:* Estabelecer uma transmissão ao vivo entre o vídeo estereoscópico das duas câmeras articuladas e a tela do smartphone usado com os óculos de realidade virtual, também com os dados de orientação do smartphone e os servos motores;

b) *Aquisição de dados do Smartphone :* Adquirir dados de orientação do magnetômetro e acelerômetro do smartphone;

c) *Controle dos Servos Motores:* Utilizar os dados de orientações obtidos para controlar dois servos;

C. Requisitos de qualidade

a) *Protótipo:* O protótipo deve ter uma estrutura sólida, seu sinal de alcance tem que ser suficiente para controlar o sistema de uma sala para outra e tem que ligar-se facilmente sem segredos ou vários procedimentos.

b) *Funcionalidade:* O sistema deve minimizar o atraso de transmissão através do servidor para o vídeo ao vivo, a aquisição de dados de smartphones e a conexão entre o robô e seu controlador. Além disso, os movimentos dos servos devem ser suaves, sem vibrações constantes.

V. DESENVOLVIMENTO

A. Articulação da Câmera

Os óculos VR concedem 3 eixos de movimentação angular Yaw, Roll e Pitch, que podem ser visualizados na Figura V-A.

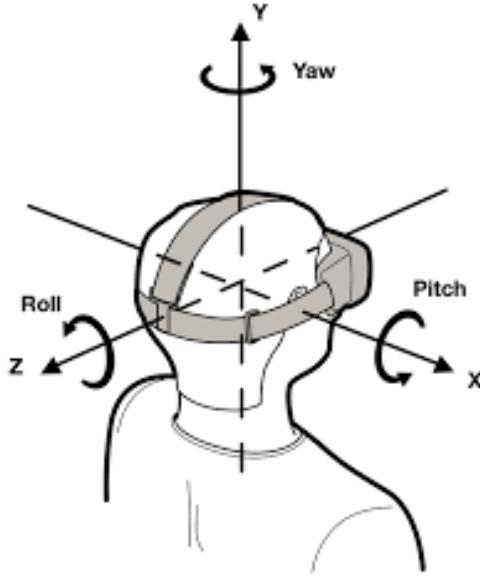


Figura 2. Coordenadas de Orientação

A câmera se movimentará de acordo com a movimentação do óculos VR, contudo em apenas dois eixos Yaw e Pitch e estes devem ser reproduzidos pela operação de dois Servos Motores. O primeiro servo e o que ficará na parte de baixo rotacionará a câmera com respeito ao ângulo Yaw e o segundo servo, que por sua vez estará em cima, rotacionará o Pitch.

Desta maneira, o controle dos servos estará condicionado a estes ângulos de ataque e para utilizá-los é possível construir uma relação entre os ângulos requisitados e o duty cycle da onda quadrada inserida nos motores.

Os servos utilizam de uma onda quadrada de frequência 50Hz ou período 20ms e o que determina a sua angulação é o ciclo de trabalho aplicado aos níveis lógicos, como ilustra a imagem V-A.

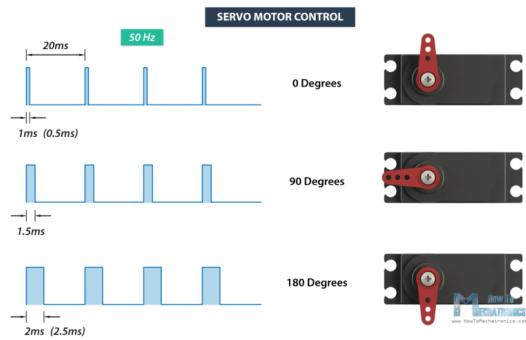


Figura 3. Funcionamento dos Servos Motores

Logo, para criarmos a fórmula basta dividir a faixa de ciclo de trabalho disponível entre 0° e 180°, que é 1000000 nanossegundos, por 180 e somar com o valor para 0°. Então o período do ciclo de trabalho (Duty Cycle) D_C em nanossegundos será:

$$D_C = 1000000 + \left(\frac{1000000}{180} \right) \cdot x^\circ \quad (1)$$

Assim, basta mudarmos o duty cycle da onda quadrada de acordo com a fórmula 1 para sintonizarmos os motores com os movimentos do óculos.

B. Onda Quadrada

As ondas quadradas utilizadas neste projeto serão PWMs (Pulse Width Modulation) gerados pela própria raspberry através do seu hardware. A escolha de utilizar uma PWM oriunda do Hardware e não do software tem sua fundamentação no fato de que possui maior qualidade e menor defasagem a medida que passa o tempo, assim permite que o produto final possa funcionar por um período maior sem prejudicar sua funcionalidade para esta parte.

Para a utilização do PWM por hardware na raspberry o grupo seguiu as instruções enunciadas na referência [9].

Depois de habilitado o uso das PWMs na raspberry, é importante lembrar que para evitar problemas o ideal é configurar primeiro o período da PWM e depois seu ciclo de trabalho. Dito isso, o tutorial permite a utilização de 2 PWMs, a pwm0 e pwm1, que estão contidas no endereço "/sys/class/pwm/pwmchip0". Para as habilitar basta escrever 0 para habilitar pwm0 e 1 para pwm1 no arquivo "export" contido na pasta "pwmchip0" com o seguinte comando:

```
1 echo 1 > /sys/class/pwm/pwmchip0/export
2 echo 0 > /sys/class/pwm/pwmchip0/export
```

Em seguida é preciso configurar as PWMs para começarem com 90°, logo deve-se escrever 200000000 nanossegundos em "period" e 1500000 em "duty_cycle":

```
1 cd /sys/class/pwm/pwmchip0/
2
3 echo 20000000 > /pwm0/period
4 echo 1500000 > /pwm0/duty_cycle
5
6 echo 20000000 > /pwm1/period
7 echo 1500000 > /pwm1/duty_cycle
```

Então é possível permitir as PWMs entrarem em funcionamento:

```
1 echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable
2 echo 1 > /sys/class/pwm/pwmchip0/pwm1/enable
```

Os sinais de PWM podem ser encontrados então na raspberry nos pinos 12 e 35 que correspondem aos nomes GPIO18 e GPIO19, respectivamente, como ilustra o página 102 da referência [10].

GPIO18	Low	PCM_CLK	S010	<reserved>	BSCSL_SDA / MISO	SPI1_CE0_N	PWM0
GPIO19	Low	PCM_FS	S011	<reserved>	BSCSL_SDL	SPI1_MISO	PWM1

Figura 4. Portas para acessar as PWMs

C. Motores DC

Os movimentos do carro serão por meio de dois motores DC, um para velocidade e outro para direção. Pelo fato do motor de tração e exigir extrema potência, será utilizado o módulo Ponte H com CI BTS7960 de 43A(Figura 5).



Figura 5. Módulo Ponte H BTS7960 43A

Enquanto o de direção utilizará o CI L293D na configuração para motor DC Bidirecional (Figura 6).

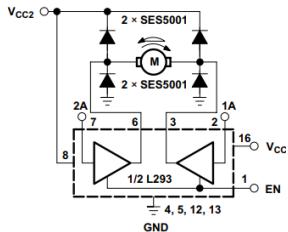


Figura 6. Ponte H Dupla L293d Motor Dc

A utilização da ponte H com CI BTS7960 para o motor de tração pode ser resumida em duas portas que indicam a orientação de rotação do motor e dois canais da habilitação do motor, um para cada sentido, que recebem PWMs para regular a potência fornecida. Por exemplo, configurando o motor para girar no sentido horário devemos usar o o pino de habilitação que realiza o movimento no sentido horário de forma a inserir um PWM nele e da mesma forma acontece para o outro sentido.

Desta maneira, para este projeto os dois pinos de habilitação estarão conectados a um mesmo PWM e bastará ajustar os níveis lógicos das portas que controlam a orientação da rotação para realizar a movimentação do Robô.

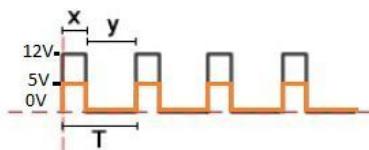


Figura 7. PWM enable ponte H

Portanto, a saída para o motor será um sinal PWM com um Duty Cycle dependente do valor analógico fornecido pelo trigger do controle do Robô e terá tensão média calculada pela seguinte formula.

$$V_{medio} = V_{max}(tensaoPonteH) * DutyCycle(\%)$$

O resultado final é o controle da velocidade do motor através de PWM.

D. Aplicativo GuearVR

O Robô VR utilizará a plataforma do Unreal Engine 4 que permite criar aplicativos para o Guear VR com uma certa facilidade e configurar o mundo 3D compatível com o aparelho da Samsung.

Para este ponto de controle foi criado um mundo com uma tela onde foi enviado uma serie de imagens que eram transmitidas pela câmera da raspberry. As imagens foram fornecidas ao aplicativo mediante ao pedido de fotos a uma alta taxa de forma a criar um vídeo. Entretanto, ainda é preciso vencer a latência entre as fotos que se aproxima de 1 segundo.

Ademais, o projeto usa servidores UDP para enviar os dados da posição e do controle para a Raspberry que foram criados apartir de BluePrints do Unreal engine.

E. Raspberry Pi Câmera

Para realizar o stream do video foi utilizado a Câmera 5mp Focal Ajustável Visão Noturna (Figura 8) para melhor qualidade de imagem em ambientes escuros.



Figura 8. Câmera 5mp Focal Ajustável Visão Noturna

A Câmera foi configurada utilizando o UV4L, software que consiste em uma série de drivers altamente configuráveis, um módulo opcional do Streaming Server que fornece uma RESTful API para desenvolvimento personalizado e várias extensões para o servidor que cooperam entre si.

O Streaming Server também fornece a interface de usuário da Web básica para que os usuários finais tentem ou usem todas as principais funcionalidades diretamente. Para eficiência máxima, cada instância do UV4L é executada como um processo de sistema único e independente que explora o hardware subjacente nativamente (sempre que possível).

Para o stream foi necessário fazer o aplicativo do Unreal Engine 4 ficar requisitando em loop uma imagem e baixa-la. Para isto foi utilizado o link:

<http://192.168.1.101:9090/stream/video.jpeg>

Contudo Obtivemos uma latência alta.

F. Transmissão Wireless

Com alta potência de transmissão, o TL-WN8200ND oferece até 10 vezes o alcance do adaptador USB de alimentação normal. Além disso, o TL-WN8200ND se conecta com a rede atravessando paredes, pisos e outros obstáculos físicos para efetivamente manter uma conexão sem fio confiável. Possuindo também taxas de transmissão de 300Mbps, ideais para tarefas que demandam muito da banda como no caso do stream de video.



Figura 9. TL-WN8200ND

G. Servidor UDP

Para a comunicação entre o aplicativo VR e a Raspberry são criados três servidores e três clientes UDP. Os servidores são instanciados na raspberry em configuração de Listening e os clientes instaciados no Unreal Engine 4 (smartphone) em configuração de Seeders.

O grupo optou pela utilização de três servidores e três clientes com 1 cliente para cada servidor, portanto três pares servidor-cliente, para dividir a aquisição de dados de controle em três principais grupos: Motor Dianteiro, Motor Traseiro e Servos Motores da Câmera.

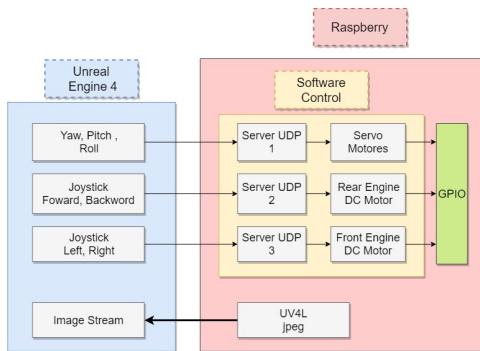


Figura 10. Diagrama Geral de Controle

1) *Servidor:* O Servidor que fará o controle da articulação da câmera precisará apenas chamar a função que altera o duty cycle das PWMs geradas por hardware toda vez que os ângulos Yaw e Pitch mudarem.

O Servidor responsável pelo Motor traseiro instanciará uma Thread para gerar um PWM por software que é enviado ao motor. Então, mediante aos valores recebidos pelo servidor, provenientes do trigger do controle do carrinho, uma variável global de duty cycle é alterada e, consequentemente, o PWM

gerado por software é atualizado. Neste mesmo servidor, é preciso analisar o valor do eixo y do joystick que indica se o robô se movimentará para frente, para trás, ou ficará parado. Por tanto, com esta informação o servidor estabelece os níveis lógicos dos enables da ponte H (BTS7960) que determinam a direção de rotação do motor e habilitam seu funcionamento.

Por fim, o servidor que controlará o motor dianteiro vai analisar os valores analógicos provenientes do eixo x do joystick e realizar a movimentação das rodas da frente, que determinam a direção, através dos pinos de direção/enable do CI L293D na configuração Motor DC Bidirecional.

2) *Cliente:* O cliente será instanciado pelo Unreal Engine 4 e fará as leituras do acelerômetro e dos botões do controle do motor. Em seguida, enviará em forma de string para os servidores que estes atuaram nos motores do Robô.

VI. MATERIAIS E CUSTOS

A. Bill of Materials

Uma análise de custos mais detalhada será feita em fases mais avançadas do projeto, porém o objetivo do grupo é manter um orçamento viável.

Material	Quan.	Custo(R\$)
Raspberry pi 3 B+	1	250,00
Câmera 5mp Visão Noturna	1	100,00
Controle Xbox	1	250,00
Servo Motores	2	22,00
Bateria 12V	1	35,00
Micro SD 32Gb	1	27,00
TL-WN8200ND	1	100,00
Ponte H 43a Bts7960	1	59,00
Pan Tilt Sg90	1	12,00
Custo Total		R\$855,00

Esta lista de materiais representa uma lista prévia dos componentes que provavelmente serão utilizados. Esta lista também não representa os componentes comprados para o projeto sendo alguns já adquiridos de outras atividades.

VII. RESULTADOS

A etapa de projeto final conseguiu entregar todos os módulos funcionando, porém com algumas limitações e detailes a serem corrigidos. Nesta seção, o grupo comentará sobre os desafios e resultados da entrega final.

A. Articulação da Câmera

A articulação da câmera utilizou dois servos motores de forma a criar dois eixos de movimentação, o Pitch e o Yaw. Entretanto, a solução final utilizada para o projeto não implementou esta estratégia, pois precisaria ser interpretada e consequentemente teria maior demanda. Dessa maneira, o grupo buscou as funções da biblioteca wiringPi que configura os PWM gerados pelo hardware da Raspberry Pi de uma forma facilitada e já compilada. Os comandos necessários para a sua configuração são os seguintes:

```

1  pinMode(Pin, PWM_OUTPUT);
2  pwmSetMode(PWM_MODE_MS);
3  pwmSetClock(clock);
  
```

```

4  pwmSetRange(rang);
5  pwmWrite(Pin, duty_cycle);

```

O resultado observado cumpriu os requisitos de movimentação e sua resposta dependeu somente da taxa de transmissão dos dados através dos servidores UDP. Ademais, os motores apresentaram rigidez na movimentação e a solução para este problema é a aplicação de uma média móvel.

B. Motor Traseiro

O motor traseiro utilizou de dois sianis PWM para o seu controle e este foram implementados por uma função da biblioteca wiringPi que instancia um PWM gerado por software. O grupo inicialmente estava desenvolvendo a lógica por trás dessas funções, mas o fato da wiringPi disponibilizar estas funções de forma compilada implicou na preferência de seu uso. Segue os comandos necessários para a sua utilização:

```

1 softPwmCreate(PwmA, minValue, pwmRange);
2 softPwmCreate(PwmB, minValue, pwmRange);
3 softPwmWrite(PwmA, duty_cycle);
4 softPwmWrite(PwmB, duty_cycle);

```

Esta implementação mostrou-se eficiente, porém o grupo encontrou um problema para mudar a orientação de rotação do motor traseiro, quando idealmente não deveria acontecer. Por isso, uma investigação sobre a utilização desta função será necessária.

C. Motor Dianteiro

Para o controle deste motor, o grupo utilizou de um módulo com a ponte H do CI L293D, portanto bastou controlar dois níveis lógicos para controlar a orientação de giro e realizar paradas bruscas, conforme ilustra a figura VII-C.

EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

Figura 11. Funções do CI L293D

Ademais, o uso do motor dianteiro é condicionado à sua lógica de controle que é determinada por um esquema físico de terminais que quando processados podem indicar a posição do motor. O grupo pensou em criar uma solução que identificasse periodicamente a orientação do motor e, então, induzisse comandos para retorná-lo para o centro, mas este processo de retorno precisa ser encerrado caso o usuário decida comandar o motor durante a sua execução.

Para solucionar este problema, utilizou-se da estrutura de tempo de expiração disponível para a função recvfrom utilizada na implementação dos servidores UDP. Toda vez que o tempo fosse expirado, o código bifurca (realiza um fork) e um processo realiza a rotina para voltar o motor pro centro

e o outro continua a verificar se o usuário enviou comandos. Esta estratégia funcionou corretamente para um servidor teste, porém por algum motivo não teve o mesmo comportamento dentro do código do projeto, de forma que a expiração do tempo não acontecia.

D. Streaming de Video

O streaming de vídeo foi considerado um sucesso, dentro das possibilidades do grupo. Vale lembrar que o mesmo foi inserido dentro de uma plataforma de Realidade Virtual e que por si só implica em diversos desafios de implementação. O grupo verificou que a latência diferiu quando o robô foi testado nos ambientes da universidade e a justificativa para isto se deve ao nível de interferência no sinal de rede e à qualidade do roteador utilizado na apresentação, que foi o próprio smartphone Samsung Galaxy S7.

O aplicativo travou o vídeo na tela do ambiente virtual e deixou a tarefa de orientação para a articulação das câmeras. Este resultado pode ser verificado durante a entrega do projeto.

E. Servidores

A utilização de três servidores UDP, consequentemente, 3 sockets mostrou-se eficiente e contribuiu para a estrutura modular do código final do projeto. Os servidores respondiam a 3 clientes diferentes e estes eram criados pelo aplicativo implementado dentro da plataforma Unreal Engine. Além disso, o grupo deu preferência para a utilização de strings para o tráfego de dados e toda essa informação foi corretamente interpretada.

F. Alimentação do Sistema

O projeto como um todo precisou alimentar diversos módulos para que o robô funcionasse de forma completa. As principais demandas vieram da Raspberry Pi, do motor dianteiro, dos servos motores, da câmera, dos circuitos regulares de tensão, das pontes H e principalmente do Motor traseiro, que em seu pico chegou à demanda de 8A. Inicialmente, o grupo utilizou de apenas uma bateria, porém durante o teste do robô com todos os módulos ligados o motor traseiro 'drivou' tanto correto que desligou a Raspberry Pi, por isso foi preciso incorporar mais uma bateria para uso exclusivo do motor traseiro.

De toda forma, o sistema de reguladores de tensão e drives de corrente foram dimensionados para aguentar os picos das baterias e fornecer as tensões corretas para os pontos de melhor eficiência dos componentes utilizados no robô.

G. Estrutura do Código

A estrutura do código final do projeto buscou organizar as funções, bibliotecas, headers e demais dependências de forma modular e com arquivos '.c' e '.h'. Para a compilação utilizou-se da ferramenta Cmake que gera um Makefile adequado e com compartilhamento de bibliotecas universal.

H. Roteamento

O roteamento desejado para o projeto esteve condicionado à utilizada da antena TL-WN8200ND, entretanto a sua configuração mostrou-se difícil e não permitiu que o grupo a utilizasse em tempo hábil. A sua presença aumentaria consideravelmente a qualidade de transmissão de dados pelos servidores UDP e a latência do streaming de vídeo.

VIII. CONSIDERAÇÕES FINAIS

Durante a implementação do projeto o grupo enfrentou 4 principais desafios: o streaming de vídeo, a utilização do ambiente de Realidade Virtual, a alimentação correta do sistema e as configurações das ferramentas e componente utilizados como a antena e o motor frontal. Frente a estes desafios o grupo conseguiu entregar um produto que carece de lapidação, ou seja, faltaram a correção e debug de detalhes de funcionamento.

É importante esclarecer que ferramentas de streaming de vídeo com baixa latência são encontradas, em sua maioria, como soluções pagas e a inserção disto em um ambiente de Realidade Virtual é um desafio e também o Estado da Arte, por isso o resultado do Police VR Robot é chamativo, pois realiza estas funcionalidades de forma ilustrativa e de certa forma funcional com uma Raspberry Pi 3.

O projeto implicou em grande desafio físico e de instrumentação eletrônica, mas também utilizou e consolidou diversas ferramentas da linguagem C, como: Sockets, Threads, Process, Shell Scripts e os pinos de GPIO, para conseguir utilizar todos os módulos propostos de forma simultânea.

REFERÊNCIAS

- [1] Robotics Business, Augmented reality vr converge robotics. Disponível em: <https://www.roboticsbusinessreview.com/ai/augmented-reality-vr-converge-robotics>. Acesso em 20 março. 2019.
- [2] Kateryna Zinchenko ,Virtual reality control of a robotic camera holder for minimally invasive surgery. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8287302&tag=1>. Acesso em 20 março. 2019.
- [3] Aniq Masood ,Stereo Pi: Portable Digital Stereo Camera. Disponível em:http://stanford.edu/class/ee367/Winter2016/Masood_Report.pdf. Acesso em 26 março. 2019.
- [4] Global Terrorism Index 2018 , Measuring the impact of terrorism. Disponível em:<http://visionofhumanity.org/app/uploads/2018/12/Global-Terrorism-Index-2018-1.pdf>. Acesso em 29 março. 2019.
- [5] optiBUSA Protection System, Advantages of Stereo Camera Technology. Disponível em: https://www.dlr.de/Portaldata/16/Resources/bahnsteeme/Flyer_en_optiBU_SA-RZ.pdf. Acesso em 29 março. 2019.
- [6] Ronak Dipakkumar, Virtual Reality – Opportunities and Challenges. Disponível em: <https://www.irjet.net/archives/V5/I1/IRJET-V5I1103.pdf>. Acesso em 29 março. 2019.
- [7] Foster Miller, TALON em: https://en.wikipedia.org/wiki/Foster-Miller_TALON. Acesso em 29 março. 2019.
- [8] Dejan, em: <https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino>. Acesso em 3 de maio. 2019.
- [9] Kellogg-Stedman, Lars. Some notes on PWM on the Raspberry Pi em: <http://blog.oddbit.com/2017/09/26/some-notes-on-pwm-on-the-raspberry-pi/>. Acesso em 3 de maio. 2019.
- [10] Raspberrypi. BCM2835 ARM Peripherals, em: <https://github.com/raspberrypi/documentation/blob/master/hardware/raspberrypi/bcm2835-BCM2835-ARM-Peripherals.pdf>. Acesso em 3 de maio. 2019.

APÊNDICE A

IMAGENS DO DESENVOLVIMENTO



Figura 12. Protótipo (1/3)



Figura 13. Protótipo (2/3)

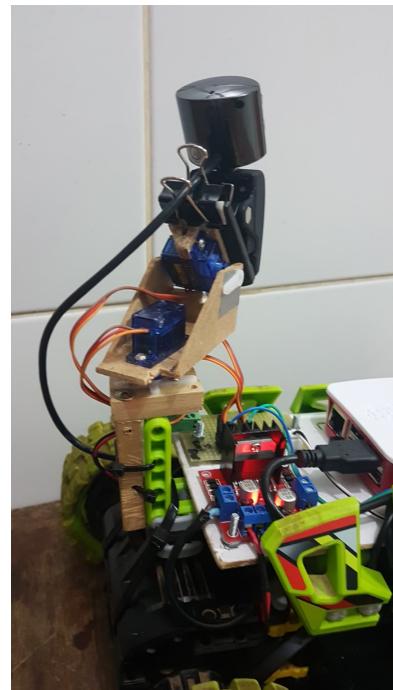


Figura 14. Protótipo (3/3).

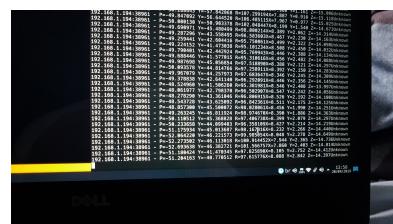


Figura 15. Comunicação UDP

APÊNDICE B PROJETOS JÁ FEITOS



Figura 16. Controlled Raspberry Pi VR Robot (1/2)



Figura 17. Controlled Raspberry Pi VR Robot (2/2)



Figura 18. Xiao R WiFi Video Robot Arm Car



Figura 19. Foster-Miller TALON SWORDS units equipped with various weaponry.