

# Um sistema de telemetria de baixo custo para veículos híbridos baseado em hardware e software abertos

José G. A. de Freitas <sup>1</sup>, Jorge Lima de O. Filho<sup>1</sup>, Álvaro V. S. Coêlho <sup>1</sup>, Leard de O. Fernandes<sup>1</sup>

<sup>1</sup>Departamento de Ciências Exatas – Universidade Estadual de Santa Cruz (UESC)  
Campus Soane Nazaré de Andrade - Rodovia Jorge Amado, km 16 -  
45.662-900 – Ilhéus – Bahia - Brasil

[golderng1@gmail.com](mailto:golderng1@gmail.com), {jlofilho,degas,lofernandes}@uesc.br

**Abstract.** *The evolution of technology increased the processing power of computers and decreasing the cost of these equipment. These facts contributed to the emergence of hybrid vehicles, which are autonomous vehicles that can be remotely controlled with the aid of an embedded video camera that transmits first-person images to a remote operator. In this work, a low-cost modular telemetry system was developed to be installed in base stations. The software makes it possible to view information from the embedded controller in the video in first person which can be adapted to any robotic system.*

**Resumo.** *A evolução da tecnologia aumentou significativamente o poder de processamento dos computadores e a redução expressiva do custo destes equipamentos, contribuindo assim, para o surgimento dos veículos híbridos, que são veículos autônomos com a possibilidade de ser controlado remotamente com auxílio de uma câmera de vídeo embarcada que transmite imagens em primeira pessoa para um operador remoto. Neste trabalho, foi desenvolvido um sistema de telemetria modular, de baixo custo, com software de código fonte aberto, para controladores de veículos híbridos que possibilita a sobreposição de informações do controlador embarcado no vídeo em primeira pessoa e que pode ser adaptado a qualquer sistema robótico.*

## 1. Introdução

A evolução da tecnologia nos últimos anos possibilitou a redução dos componentes eletrônicos aumentando significativamente o poder de processamento dos computadores e a redução expressiva do custo destes equipamentos. Com a popularização destes componentes começou a surgir no mercado uma nova geração de veículos com os mais diversos propósitos (transporte, entretenimento, monitoramento, etc), que não dependem estritamente do controle humano para funcionamento. Estes veículos, em geral, podem ser autônomos, semiautônomos ou controlados remotamente [Garcia et al. 2007].

Os veículos semiautônomos, também conhecidos por veículos híbridos, podem ser controlados remotamente com auxílio de uma câmera de vídeo embarcada, que transmitem imagens em primeira pessoa para um operador remoto [Limsoonthrakul et al.

2008]. O principal objetivo dos veículos híbridos é combinar uma série de sistemas de controle (sensores e atuadores) para perceber o ambiente ao seu redor, assim ele pode determinar as melhores opções de ação e executá-las de forma confiável. Todos estes sensores e atuadores, geralmente, são controlados por um computador embarcado, também conhecido como controlador, que é responsável por executar os algoritmos de percepção do ambiente e controle do veículo.

Nos veículos híbridos, quando necessário, um operador remoto pode intervir no veículo e tomar o seu controle temporariamente. Uma característica dos veículos híbridos é a necessidade do uso de uma estação base de controle, responsável por receber todas as informações (sensores, vídeo, etc) e por enviar as informações de controle ao veículo. O procedimento de envio e recebimento destas informações é conhecido como telemetria.

Atualmente existem diversos sistemas de telemetria proprietários como Itelemetry [Control 2019], XPAD2 [Devices 2019] e Eagle Tree [Tree 2019] que possibilitam a inclusão da função de telemetria nos veículos. Contudo, estes sistemas possuem algumas limitações: alto custo, complexa integração entre sistemas, longa curva de aprendizado, impossibilidade de adicionar novos sensores, hardware fechado (poucas informações), etc. Também existem alguns sistemas abertos de telemetria [OPLINK 2015; Robotics 2014], porém, para integrar esses sistemas aos controladores é preciso criar uma interface entre o controlador e o transmissor de dados, o que restringe bastante o uso destes sistemas por pessoas leigas. Além disso, para visualizar as informações na estação base é preciso uma tela dedicada para exibição dos dados do veículo. O ideal seria que o operador pudesse se concentrar na tela do vídeo sem se distrair com as informações do veículo remoto.

Existem veículos no mercado que exibem no raio de visão do piloto informações do equipamento, como por exemplo, velocidade, altitude etc. Estes sistemas são conhecidos como *Head up display* [Yung-Ching, L. et al. 2004]. Outro mecanismo que é usado para mesclar informações do veículo com o vídeo em primeira pessoa é o *On-screen display* (OSD), que é uma imagem sobreposta a um vídeo de tela, bastante usado em televisores para exibir informações como volume, canal etc. Normalmente, este mecanismo é instalado no veículo e possui duas interfaces: uma que recebe as informações dos sensores atrelados ao controlador e outra que recebe um sinal de vídeo. Assim, as informações são sobrepostas no vídeo e logo em seguida enviado para o transmissor de vídeo, que é repassado para a estação base. Apesar do baixo custo destes sistemas, eles possuem algumas desvantagens, como por exemplo: a impossibilidade de customizar quais informações dos sensores serão exibidas na estação base, a necessidade de integração com o controlador, e a sobrecarga na transmissão de dados de vídeo entre o veículo e a estação base. Há no mercado uma série de sistemas OSDs como Minim OSD [3D Robotics 2020] e EzOSD [Immersion RC 2020], porém, eles são dependentes do controlador, ou seja, a interface de conexão OSD do controlador precisa ser compatível com a interface do sistema OSD.

Diante do cenário exposto, foi projetado e desenvolvido neste trabalho um sistema de telemetria modular, de baixo custo, com software de código fonte aberto, para controladores de hardware aberto de veículos híbridos que possibilita a sobreposição de informações do controlador no vídeo em primeira pessoa e que pode ser adaptado a qualquer sistema robótico. Os testes realizados com o sistema de telemetria verificou-se facilitar o trabalho do usuário remoto, barateando o custo operacional do operador.

## 2. Fundamentação Teórica

Esta seção apresenta as principais tecnologias e protocolos utilizados para no desenvolvimento do sistema de telemetria proposto.

### 2.1. MAVLink

É um protocolo de transmissão e recepção de mensagens que foi criado com o propósito de facilitar a comunicação entre controladores de veículos não tripulados e suas estações bases [Mavlink 2020]. O MAVLink utiliza *streaming* de dados para a transmissão de mensagens entre hardware e software utilizando *bytes* contendo informações com o objetivo de facilitar a obtenção dessas informações de telemetria pelo usuário. As mensagens são definidas dentro de arquivos XML que definem um conjunto de mensagens suportadas por um determinado sistema MAVLink.

No programa implementado foi escolhido um conjunto de mensagens (altitude, velocidade, nível de bateria, posição geográfica, etc) relevantes para o controle e o status do veículo e que foi definido no arquivo `common.xml`. Uma vez definido o arquivo `common.xml`, foi utilizado um conjunto de ferramentas do MAVLink para gerar a biblioteca utilizada no programa desenvolvido que foi codificado na linguagem de programação Python utilizando a biblioteca Dronekit.

### 2.2. Dronekit

A biblioteca Dronekit [3D Robotics, inc 2020] facilita o desenvolvimento de aplicações que necessitam realizar a comunicação com os controladores presentes nos veículos remotos. As mensagens são trocadas utilizando o MAVLink como protocolo de comunicação. Foi utilizada para o desenvolvimento do software o `dronekit-python`, que permite o uso de chamadas API para obtenção das informações de telemetria no controlador presente no veículo. Foi preciso instalar o Dronekit no controlador embarcado no veículo e importar a biblioteca no software de telemetria.

### 2.3. OpenCV

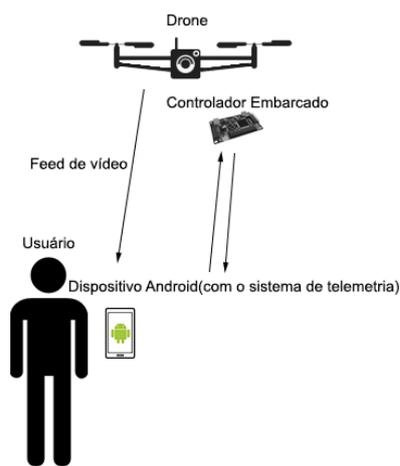
O OpenCV [OpenCV 2020] é uma biblioteca multiplataforma para o desenvolvimento de aplicativos na área de visão computacional. O OpenCV possui módulos de processamento de imagens e vídeo, estrutura de dados, álgebra linear, GUI (*Graphical User Interface* - Interface Gráfica do Usuário) básica com sistema de janelas independentes, controle de mouse e teclado, além de mais de 350 algoritmos de Visão computacional. O seu processamento de imagens é em tempo real.

Esta biblioteca foi escolhida para o uso no sistema de telemetria pela facilidade em integração com a linguagem de programação Python e pela facilidade de conversão de imagens em matrizes ou outros tipos de dados que podem ser enviados por servidores agilizando o transporte de informações. O OpenCV foi utilizado neste projeto para tratar a imagem da câmera integrada ao veículo remoto e inserir as informações de telemetria sobre o vídeo para visualização do operador.

## 3. Sistema de telemetria

Nesta seção é descrito o contexto onde é usado o sistema de telemetria desenvolvido neste trabalho como também os principais componentes que o formam.

A Figura 1 ilustra um usuário de posse de um dispositivo móvel conectado a um controlador embarcado num veículo remoto (*drone*). O sistema de telemetria foi codificado utilizando a linguagem de programação Python e a biblioteca Kivy [Kivy 2020] o que possibilita que o sistema seja instalado em qualquer plataforma. Neste cenário o usuário tem o vídeo em tempo real, em primeira pessoa, do veículo móvel com as informações vitais de telemetria sobrepostas no vídeo. As informações de telemetria podem ser configuradas pelo usuário e vão ser dependentes do controlador. As principais informações de telemetria são responsáveis em manter a operação do veículo com segurança, por exemplo, altitude, nível da bateria, posição geográfica, etc.

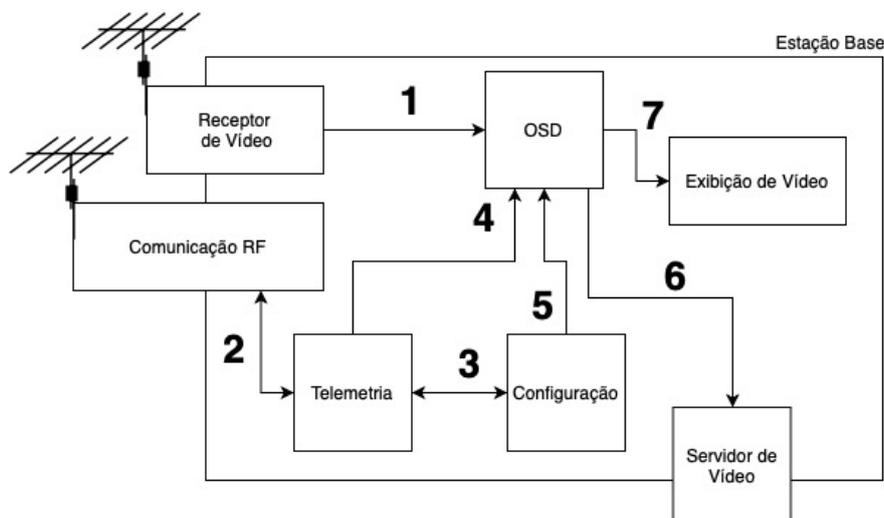


**Figura 1. Cenário de uso do sistema de telemetria.**

O sistema de telemetria desenvolvido é formado por duas partes: a primeira parte é composta pela estação base que o usuário observa todas as informações de telemetria numa tela com o vídeo em tempo real. A segunda parte fica no veículo remoto, que é composto pelo controlador e pelo sistema de vídeo.

A Figura 2 ilustra a estação base e seus principais componentes. A estação base é conectada em tempo real com o veículo remoto por dois canais de comunicação. Um canal é responsável pelo vídeo em tempo real e o outro canal recebe todas as informações de telemetria, também em tempo real, utilizando o protocolo MAVLink sobre um canal de rádio frequência. A estação base é composta por 7 módulos principais: o receptor de vídeo, o módulo de comunicação RF, o módulo de telemetria, o OSD, o módulo responsável pela configuração, o módulo de exibição de vídeo e por fim o servidor de vídeo. O receptor de vídeo envia o vídeo recebido em tempo real do veículo remoto para o sistema OSD (1). O OSD é responsável por sobrepor as informações de telemetria obtidas pelo módulo de telemetria (4) na fonte de vídeo. Para isso, ele consulta também o módulo de configuração (5) em que o usuário define quais informações que ele quer que seja exibida na tela. O módulo de telemetria também consulta o módulo de configuração (3) para saber quais informações ele deve solicitar ao controlador através do módulo de comunicação de RF (2). O módulo de comunicação de RF solicita ao veículo remoto as informações requisitadas pelo módulo de telemetria em intervalos de tempo regulares. O módulo de exibição de vídeo exibe o vídeo em tempo real já com todas as informações de telemetria que foram sobrepostas pelo módulo OSD (7) e por fim

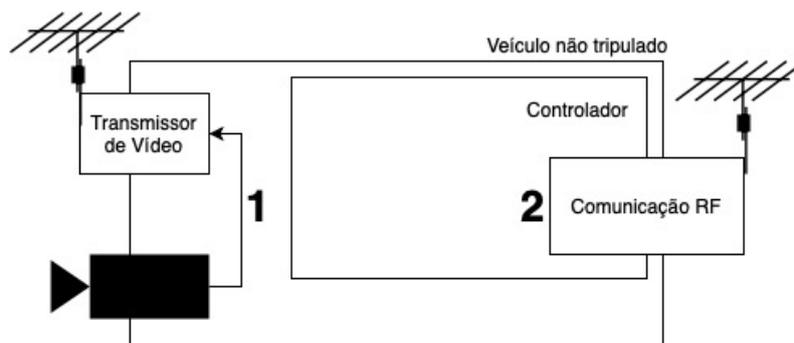
foi implementado um servidor de vídeo que permite que clientes se conecte ao sistema de telemetria através de uma conexão IP e acesse o vídeo em tempo real de qualquer lugar do globo.



**Figura 2. Principais componentes do sistema de telemetria na estação base.**

O sistema implementado na estação base possibilita a exibição de vídeo do veículo remoto em tempo real numa tela com as informações de telemetria sobrepostas. O vídeo poderá também ser disponibilizado remotamente através de uma conexão IP ao servidor de vídeo.

A Figura 3 ilustra o veículo não tripulado com seus principais componentes. O veículo possui uma câmera que transmite o vídeo em tempo real para a estação base (1) e o controlador do veículo que transmite as informações de telemetria para a estação base (2) quando requisitadas através do módulo de comunicação RF. No veículo remoto, limitou-se apenas em instalar a biblioteca dronekit e ajustar os parâmetros de comunicação no controlador, além de configurar a fonte de vídeo a ser enviada para estação base através do transmissor de vídeo.



**Figura 3. Principais componentes do sistema de telemetria no veículo remoto.**

## 4. Resultados

Nesta seção são apresentadas algumas telas do sistema de telemetria, o *testbed* montado, para testar a viabilidade do sistema e um experimento que contabiliza a perda de pacotes de dados MAVLink, que são informações de telemetria em função da distância entre o veículo remoto e a estação base.



**Figura 4. Tela inicial do sistema de telemetria.**

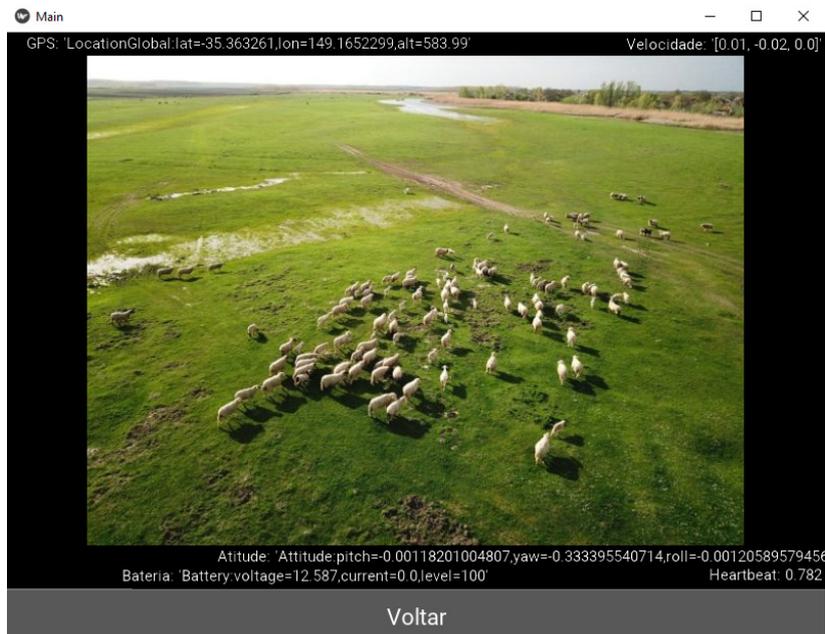
A Figura 4 ilustra a tela inicial do sistema. O usuário inicialmente precisa acessar as configurações e escolher um controlador dentre os disponíveis e se conectar. Após a conexão ser estabelecida com o controlador o usuário pode ter acesso ao vídeo com as informações de telemetria.

A Figura 5 exibe a tela de configurações do sistema de telemetria. O usuário tem três opções principais: um botão de configuração de controlador, um botão de configuração do servidor de vídeo e uma área com todos os parâmetros (informações de telemetria) disponíveis que são previamente consultados após a conexão com o controlador e que ficam disponíveis para o usuário. A configuração do controlador possibilita o usuário escolher o tipo de conexão, o controlador disponível e a velocidade para fins de conexão. A opção de configurações do servidor possibilita ao usuário escolher a porta de conexão, a quantidade de conexões simultâneas que o sistema vai aceitar, um botão de iniciar e outro de parar o servidor de vídeo. E por fim aparecem todos os parâmetros que o usuário pode escolher para que seja exibido sobre o vídeo.



**Figura 5. Tela de configurações após conexão com o controlador.**

A Figura 6 ilustra a tela câmera exibindo o vídeo e as informações de telemetria configuradas pelo usuário em tempo real. O usuário pode posicionar os parâmetros de telemetria em qualquer disposição na tela de acordo com sua necessidade.



**Figura 6. Tela de câmera exibindo as informações de telemetria em tempo real.**

## Testbed

Para a montagem do testbed foi construído um veículo tipo *drone* com 4 motores utilizando um controlador pixhawk [Pixhawk 2020] com barômetro, GPS, transceptor de RF configurado na frequência 433 MHz. Além disso, foi instalado no *drone* uma câmera conectada a um transmissor de vídeo configurado na frequência de 5.8 GHz [Immersion 2020]. No lado da estação base, foi instalado o sistema de telemetria num notebook rodando o Windows 10. Um transceptor de RF [HKPilot 2020] configurado na frequência de 433 MHz foi utilizado na porta USB do notebook para se conectar ao *drone* e assim

obter as informações de telemetria do controlador. Por fim foi usado um receptor de vídeo para obter as imagens em tempo real da câmera do *drone*.

O Quadro 1 exibe a porcentagem de perda de pacotes MAVLink em duas situações, com linha de visada, onde a antena da estação base estava apontada diretamente para a antena do veículo remoto sem nenhum obstáculo entre elas e sem linha de visada, onde entre as duas antenas tinha algum tipo de obstáculo entre elas. O experimento limitou-se a afastar o *drone* da estação base por um minuto e contar os pacotes MAVLink perdidos em diferentes distâncias. Com linha de visada quase não houve perdas, ao contrário da existência de obstáculos entre as antenas que houve 6% de perdas com 100 metros de distância e 20% de perdas com 200 metros de distância. A qualidade de vídeo não sofreu alterações significativas com a distância máxima de 200 metros que pudesse comprometer a controlabilidade do veículo.

O sistema de telemetria funcionou como esperado, atendendo aos requisitos de usabilidade especificados na fase de projeto. O sistema desenvolvido permite a um operador controlar o veículo remoto corretamente com base em informações de telemetria sobrepostas no vídeo da estação base. Por ser um sistema de código fonte aberto, todo o código foi disponibilizado na web e pode ser baixado no site do Github [Freitas, J.G.A 2020].

**Quadro 1. Perdas de pacotes (%) de dados de telemetria em função da distância entre o *drone* e a estação base.**

	Com linha de visada	Sem linha de visada
5 metros	0	0
50 metros	0	4%
100 metros	0	6%
200 metros	2%	20%

## 6. Conclusões e trabalhos futuros

Neste trabalho foi apresentado um sistema de telemetria de baixo custo, de código fonte aberto e compatível com a maioria dos controladores disponíveis no mercado. Um *testbed* foi montado e experimentos foram realizados para testar o sistema num veículo real com baixa perda de pacotes em diferentes distâncias. Vários desafios técnicos foram superados no decorrer do desenvolvimento deste projeto, como por exemplo, como sobrepor informações vinda do controlador no vídeo, como compactar o vídeo para ser disponibilizado para os clientes do sistema e como montar um *drone* para fins de pesquisa.

Com os resultados obtidos neste trabalho é possível afirmar que o sistema é eficaz para auxiliar no controle de veículos não tripulados, fazendo com que o usuário fique focado no que está vendo em primeira pessoa e ao mesmo tempo confiando nas informações de telemetria que chega na estação base em tempo real.

Foi pensada várias melhorias para uma futura versão do sistema de telemetria como: interface específica para o controle do veículo remoto na estação

base, criador de missões baseadas em aprendizagem de máquina, análise de imagens recebidas para fins específicos (ex: contagem de gado no pasto, análise em tempo real de tráfego de veículo, etc). Além disso, está previsto testes de usabilidade do sistema para melhoria de interface homem máquina.

## Referências

- Garcia, E., Jimenez, M. A., De Santos, P. G. and Armada, M. (2007). The evolution of robotics research. *IEEE Robotics and Automation Magazine*, v. 14, n. 1, p. 90– 103.
- Limsoonthrakul, S., Dailey, M. N., Srisupundit, M., Tongphu, S. and Parnichkun, M. (2008). A modular system architecture for autonomous robots based on blackboard and publish-subscribe mechanisms. In 2008 IEEE International Conference on Robotics and Biomimetics, ROBIO 2008.
- Control, I. R. (2019). ITelemetry Dongle. Disponível em: <http://www.immersionrc.com/fpv-products/itelemetry-dongle/> Acessado em 29 de Setembro de 2019.
- Devices, D. M. (2019). XPAD2. Disponível em: [http://www.xlrs.eu/xlrs\\_d2\\_xpad2.htm](http://www.xlrs.eu/xlrs_d2_xpad2.htm) Acessado em: 13 de Setembro de 2019.
- Tree, E. (2019). User Manual for the Seagull Wireless Dashboard Telemetry and Data Recorder Systems Pro, Flight, Glide, Boat. Disponível em: <http://www.eagletreesystems.com/Manuals/Pro, Glide, Flight and Boat Seagull and Data Recorder Instruction Manual.pdf>. Acessado em 5 de Outubro de 2019.
- OPLINK (2015). OPLink Telemetry System. Disponível em: [http://opwiki.readthedocs.io/en/latest/user\\_manual/oplink/cc.html#hardware-connections](http://opwiki.readthedocs.io/en/latest/user_manual/oplink/cc.html#hardware-connections) Acessado em 1 de Setembro de 2019.
- Robotics, 3D (2014). 3DR RADIO. Disponível em: <https://3dr.com/wp-content/uploads/2013/10/3DR-Radio-V2-doc1.pdf> Acessado em 15 de Agosto de 2016.
- Yung-Ching, L. Ming-Hui, W. (2004). Comparison of head-up display (HUD) vs. head-down display (HDD): driving performance of commercial vehicle operators in Taiwan. In Elsevier International Journal of Human-Computer Studies, v.61, n. 5, p.679--697
- 3D Robotics (2020). MINIM OSD. Disponível em: <https://hobbywireless.com/flight-controllers-accessories-c-146-185/mini-osd-module-for-apm-pix-pixhawk-flight-control-board-p-4741.html?zenid=202f153b343284d9d21d7b1423fb0a3f>. Acessado em: 01 de Março de 2020.
- Immersion RC (2020). EzOSD On Screen Display. Disponível em: <http://www.immersionrc.com/fpv-products/ezosd/>. Acessado em 02 de Março de 2020.
- Mavlink (2020). Guia de desenvolvimento Mavlink. Disponível em: <https://mavlink.io/en/>. Acessado em 10 de Março de 2020.
- 3D Robotics, inc (2020). Dronekit python. Disponível em: <https://github.com/dronekit/dronekit-python>. Acessado em 12 de Março de 2020.

- OpenCV (2020). Documentação da biblioteca Open Source Computer Vision. Disponível em: <https://docs.opencv.org/4.2.0/d1/dfb/intro.html> . Acessado em 7 de Março de 2020.
- Kivy (2020). Getting started with kivy. Disponível em: <https://kivy.org/doc/stable/gettingstarted/intro.html>. Acessado em 1 de Março de 2020.
- Pixhawk (2020). HKPilot32 Flight Controller. Disponível em: [http://docs.px4.io/v1.9.0/en/flight\\_controller/HKPilot32.html](http://docs.px4.io/v1.9.0/en/flight_controller/HKPilot32.html). Acessado em 5 de Janeiro de 2020.
- Immersion (2020). Transmissor de vídeo immersion 5.8 Ghz tramp HV. Disponível em: <https://www.immersionrc.com/fpv-products/tramp-hv-5-8ghz-video-transmitter/>. Acessado em 3 de Março de 2020.
- HKPilot (2020). HK Pilot Telemetry Radio. Disponível em: [http://docs.px4.io/v1.9.0/en/telemetry/hkpilot\\_sik\\_radio.html](http://docs.px4.io/v1.9.0/en/telemetry/hkpilot_sik_radio.html). Acessado em 2 de Fevereiro de 2020.
- Freitas, J.G.A. (2020). Um sistema de telemetria de baixo custo. Disponível em: <https://github.com/goldern/Telemetria>. Acessado em 10 de Março de 2020.