

UNIVERSIDADE DE PASSO FUNDO

Daniel Camargo Zuselski

CONTROLE DE VANT COM GPS.

Passo Fundo

2024

Daniel Camargo Zuselski

## CONTROLE DE VANT COM GPS.

Trabalho apresentado ao curso de Engenharia Elétrica, do Instituto de Tecnologia, da Universidade de Passo Fundo, como requisito parcial para obtenção do grau de Engenheiro Eletricista, sob orientação do professor Dr. Adriano Luís Toazza.

Passo Fundo

2024

Daniel Camargo Zuselski

**Controle de VANT com GPS.**

Trabalho apresentado ao curso de Engenharia Elétrica, do Instituto de Tecnologia, da Universidade de Passo Fundo, como requisito parcial para obtenção do grau de Engenheiro Eletricista, sob orientação do professor Dr. Adriano Luís Toazza.

Aprovado em \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_.

**BANCA EXAMINADORA**

---

Prof. Dr. Orientador Adriano Luís Toazza - UPF

---

Prof. M Sc. Edson Santos Acco - UPF

Este trabalho é dedicado aos meus pais Adélia e Carlos (em memória),  
por todo o esforço que fizeram para tornar seus filhos boas pessoas.

## AGRADECIMENTOS

Agradeço a Deus primeiramente por toda a sorte que tive na vida, por ter me dado força e calma para nunca desistir e por sempre continuar apesar das dificuldades.

Agradeço imensamente aos meus pais Adélia e Carlos por toda sua luta para criar seus filhos da melhor maneira possível, por serem pais amorosos e dedicados e nós ensinarem a sermos boas pessoas a mim e a minha irmã, quero agradecer também a minha irmã Deisi, que foi minha parceira em muitas batalhas da vida, sempre me cuidando.

Aos professores que sempre fizeram o seu melhor para nos transmitir seu conhecimento, aos colegas que sempre dispuseram seu tempo a ajudar mesmo tendo muitas outras coisas para fazer.

A minha esposa Verônica, quero fazer um agradecimento especial, por ser uma pessoa incrível, otimista, entusiasmada e proativa, por ter me ajudado a sair da zona de inércia em que eu me encontrava e me incentivar a buscar meus sonhos, mesmo que fosse um passinho por vez, de passinho em passinho quando me dei conta já estava fazendo TCC, só ela sabe a emoção que senti quando peguei o meu aceite de ingresso na UPF, por sempre me apoiar nas horas difíceis, e nos infundáveis finais de semana estudando e fazendo trabalhos, nas horas de tristeza, principalmente após as provas, e me escutar quando eu fazia algo que eu considerava incrível na faculdade e ficava tagarelando sobre isso, por ser esse apoio e por muitas vezes ser crítica dos meus trabalhos, sempre acreditando que eu podia mais, se eu cheguei até aqui é por que você estava ao meu lado.

“Os sonhos têm o poder de nos levar a patamares impensáveis. Quem dera fossemos todos sonhadores!”

Augusto Cury

## RESUMO

No cenário contemporâneo, os Veículos Aéreos Não Tripulados (VANTs), popularmente conhecidos como drones, têm se destacado não apenas como instrumentos de entretenimento, mas como ferramentas multifuncionais capazes de redefinir a interação entre tecnologia e sociedade. No âmbito deste projeto de Trabalho de Conclusão de Curso (TCC), propõe-se a montagem de um drone do tipo *do it yourself* (DIY) na tradução literal é faça você mesmo, elevando a experiência de construção à exploração de um sistema inovador de integração. A peça central desta proposta reside no desenvolvimento de um sistema baseado em GPS do tipo *follow me*, projetado para acompanhar um objeto móvel predefinido. Este avanço tecnológico visa não apenas aprimorar a autonomia dos VANTs, mas também abrir portas para uma série de aplicações práticas em diferentes cenários. A utilização do sistema de posicionamento global (GPS) como ferramenta fundamental para o acompanhamento do objeto em movimento representa uma ponte entre a tecnologia DIY e a vanguarda dos sistemas autônomos. O objetivo principal é explorar as capacidades desses drones, não apenas como brinquedos aéreos, mas como agentes autônomos capazes de desempenhar funções específicas. O potencial deste projeto é vasto, abrangendo áreas como entretenimento, monitoramento e vigilância. A capacidade do drone em seguir o objeto alvo não apenas aprimora a experiência do usuário, mas também promove inúmeras possibilidades de aplicação prática. Desde transmissões aéreas inovadoras em eventos esportivos até operações de vigilância estratégica, um drone com capacidade de seguimento GPS representa um avanço significativo na convergência entre eletrônica, automação e controle.

Palavras-Chave: Drone, GPS, seguidor, autonomia, rastreamento, controle.

## **ABSTRACT**

In the contemporary landscape, Unmanned Aerial Vehicles (UAVs), commonly known as drones, have emerged not merely as entertainment devices but as multifunctional tools capable of reshaping the interaction between technology and society. Within the scope of this Final Graduation Project (TCC), the proposal involves the assembly of a do-it-yourself (DIY) drone, taking the construction experience to a new level by exploring an innovative integration system. The core of this proposal lies in the development of a GPS-based tracking system designed to follow a predefined mobile object. This technological advancement aims not only to enhance the autonomy of UAVs but also to open doors to practical applications in various scenarios. The use of the Global Positioning System (GPS) as a fundamental tool for tracking a moving object serves as a bridge between DIY technology and the forefront of autonomous systems. The primary objective is to explore the capabilities of these drones not just as aerial toys but as autonomous agents capable of performing specific functions. The potential of this project is extensive, spanning areas such as entertainment, monitoring, and surveillance. The drone's ability to track a target object not only enhances the user experience but also fosters numerous possibilities for practical applications. From innovative aerial broadcasts at sports events to strategic surveillance operations, the UAV with GPS tracking capability represents a significant advance in the convergence of electronics, automation, and control.

Keywords: Drone, GPS, tracker, autonomy, tracking, control.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Parafuso Aéreo.	16
Figura 2 – 14-BIS de Santos Dumont.	17
Figura 3 – Mercado Global de drone .	17
Gráfico 1 – Solicitações de vôos no espaço aéreo no Brasil.	18
Quadro 1 – Componentes essenciais e componentes opcionais.	19
Figura 4 – Componentes de um Drone.	20
Figura 5 – Drone DIY.	21
Figura 6 – Rotação das hélices.	22
Figura 7 – Controladoras Pixhawk e KK2.	23
Figura 8 – Planos orbitais dos satélites de GPS.	24
Figura 9 – Triangulação a partir dos satélites é à base do sistema do GPS.	24
Figura 10 – Receptor GPS Digital Genérico Diagrama de Blocos.	25
Quadro 2 – Formatação NMEA.	26
Quadro 3 – Significado da sentença.	27
Figura 11 – Diagrama Sistemas de comunicação.	28
Figura 12 – Piconet e Scatternet.	30
Figura 13 – Módulo Bluetooth Low Energy - HC-08.	31
Quadro 4 - Principais Protocolos Wi-Fi.	32
Figura 14 – Módulo Wi-Fi serial ESP8266.	32
Figura 15 – Arquitetura de rede LoRaWAN.	33
Figura 16 - Módulo LoRa.	34
Figura 17 – Alcance e largura de banda.	34
Figura 18 – ESP32	36
Figura 19 – Diagrama de blocos ESP32.	36
Figura 20 – Esquemático genérico.	38
Figura 21 – Leitura de GPS.	39
Figura 22 – Trajetória vetorial.	40
Figura 23 – Simulação de GPS.	41
Figura 24 – Estrutura de simulação.	41
Figura 25 – Placa LILYGO.	42
Figura 26 – Mensagens NMEA	42

Figura 27 – Bateria.	44
Figura 28 – Controladora Pixhawk.	45
Figura 29 – Entradas da controladora Pixhawk.	46
Figura 30 – LED's da controladora Pixhawk.	47
Figura 31 – M8N.	48
Figura 32 – Controle FS-I6.	49
Figura 33 – PPM.	50
Figura 34 – Receiver FS-IA10B.	51
Figura 35 – Frame.	51
Figura 36 – Motor.	53
Figura 37 – ESC.	53
Figura 38 – ESC controle de acionamento.	54
Figura 39 – Telemetria.	55
Figura 40 – BEC.	55
Figura 41 – Tela de gravação do firmware.	56
Figura 42 – Mission Planner tela inicial.	57
Figura 43 – Tela de calibração do radio.	58
Figura 44 – Comunicação.	59
Figura 45 – Caixa de dialogo.	60
Figura 46 – Void loop.	60
Figura 47 – Dados da NMEA.	61
Figura 48 – Envio do destino.	62
Figura 49 – Envio do destino erros.	63
Figura 50 – Frame desmontado.	64
Figura 51 – Frame montado.	65
Figura 52 – Testes.	66
Figura 53 – Drone.	67
Figura 54 – Celular e telemetria.	68
Figura 55 – Drone em teste.	68
Figura 56 – Drone em teste 2.	69
Figura 57 – Drone em teste 3.	69

## **LISTA DE TABELAS**

Tabela 1 – Comparação entre as redes sem fio	35
Tabela 2 – Gastos	70

## LISTA DE SIGLAS

VANT - Veículo Aéreo Não Tripulado

UAV - *Unmanned Aerial Vehicles*

VARP - veículo Aéreo Remotamente Pilotado

DIY - *Do It Yourself* (faça você mesmo)

RSSI - *Received Signal Strength Indicator* (Indicador de intensidade do sinal recebido)

GPS – Sistema de Posicionamento Global

DECEA - Departamento de Controle do Espaço Aéreo

ASCII – *American Standard Code for Information Interchange*

CR - *carriage return*

LF - *line feed*

FPV - *First Person View* (Visão em primeira pessoa)

A/D - Analógico Para Digital

D/A – Digital Para Analógico

ISM – *Industrial Scientific and Medical*

WPAN - *Wireless Personal Area Networks*

IEEE – *Institute of Electrical and Electronic Engineers*

MAC - *Media Access Control*

IoT - *Internet of Things*

BLE - *Bluetooth Low Energy*

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>13</b>
1.1 CONTEXTO.....	14
1.2 OBJETIVO GERAL.....	14
1.3 OBJETIVOS ESPECÍFICOS .....	14
1.4 JUSTIFICATIVA .....	15
1.5 ESTRUTURA DO TRABALHO .....	15
<b>2 REVISÃO DA LITERATURA .....</b>	<b>16</b>
2.1 HISTÓRIA DO VANT.....	16
<b>2.1.1 VANT e drone .....</b>	<b>18</b>
<b>2.1.2 Estrutura .....</b>	<b>19</b>
<b>2.1.3 Funcionamento .....</b>	<b>21</b>
2.2 POSICIONAMENTO.....	23
2.3 COMUNICAÇÃO .....	27
<b>2.3.1 Bluetooth .....</b>	<b>29</b>
<b>2.3.2 Wi-Fi .....</b>	<b>32</b>
<b>2.3.3 LoRa.....</b>	<b>33</b>
2.4 DADOS .....	35
<b>2.4.1 ESP32.....</b>	<b>36</b>
<b>3 DESENVOLVIMENTO DO PROJETO.....</b>	<b>37</b>
3.1 ESTRUTURA MECÂNICA/PROTÓTIPO .....	44
3.2 HARDWARE .....	44
<b>3.2.1 Bateria .....</b>	<b>44</b>
<b>3.2.2 Controladora.....</b>	<b>46</b>
<b>3.2.3 GPS .....</b>	<b>49</b>
<b>3.2.4 Radio controle.....</b>	<b>50</b>

<b>3.2.5 Frame</b> .....	<b>53</b>
<b>3.2.6 Motores</b> .....	<b>53</b>
<b>3.2.7 ESC</b> .....	<b>54</b>
<b>3.2.8 Telemetria</b> .....	<b>56</b>
<b>3.2.9 BEC</b> .....	<b>56</b>
<b>3.3 FIRMWARE</b> .....	<b>57</b>
<b>3.4 SOFTWARE</b> .....	<b>58</b>
<b>3.4.1 Follow me</b> .....	<b>61</b>
<b>4 RESULTADOS E DISCUSSÃO</b> .....	<b>66</b>
<b>5 CONSIDERAÇÕES FINAIS</b> .....	<b>73</b>
<b>REFERÊNCIAS</b> .....	<b>75</b>
<b>APÊNDICE A – CRONOGRAMA SIMPLIFICADO</b> .....	<b>Erro! Indicador não definido.</b>
<b>ANEXO A – CÓDIGO FOLLOW ME</b> .....	<b>79</b>

## 1 INTRODUÇÃO

No mundo em constante evolução, os drones ou veículos aéreos não tripulados (VANTs), tem ganho cada vez mais espaço e destaque devido a sua versatilidade. A indústria dos VANTs alcançou feitos notáveis, movimentou 300 milhões no Brasil em 2018, apenas um ano após sua regulamentação, o setor cresce exponencialmente, trazendo consigo novas aplicações e experiências que fazem os VANTs cada vez mais tecnológicos e sofisticados. (ESTADO DE MINAS, 2018).

Segundo SILVA et al. (2015) os VANTs tem sido um campo de pesquisa e exploração crescente em diversas áreas que vão desde o entretenimento, como gravação de vídeos e fotografias aéreas, passando pelo melhor aproveitando do espaço aéreo ainda subutilizado, até o campo, onde na agricultura já realizam diversas tarefas como a aplicação de pesticidas, fertilizantes e colheita dos frutos.

No campo de coleta de dados, estão sendo utilizados para mapeamento de terrenos com a criação de modelos 3D do relevo, podem também fazer o monitoramento do avanço do desmatamento, identificação de ocupações irregulares, etc. (LEMMENS, 2015).

Com essa gama bem variada de aplicações os VANTs vêm ganhando cada vez mais protagonismo no cenário moderno, a integração de avançados chips de processamento e códigos de processamento de imagem podem ser utilizados para reconhecimento facial, detecção de calor em paredes e até mesmo detecção de ferrugens em partes estruturais de plantas (RANGEL et al. 2022).

No Brasil, os VANTs já foram empregados em algumas operações, como a Ágata, das Forças Armadas na faixa de fronteira da Amazônia. Seu uso visava combater crimes de fronteira, como contrabando e tráfico de drogas, dentre outros. Também estiveram presentes na conferência Rio+20, como equipamentos de vigilância. Sua utilização no monitoramento aéreo coloca os policias em vantagem tática, facilitando as tomadas de decisões (AGUIAR, 2017).

A integração de VANTs com outros aparelhos eletrônicos ainda abre outras possibilidades como o acompanhamento de atletas, segundo FRAĆKIEWICZIN (2023), por exemplo, os drones podem ser usados para fornecer informações em tempo real sobre o desempenho de um atleta, como velocidade, distância e frequência cardíaca. Esses dados podem ser usados para ajustar o regime de treinamento de um atleta e garantir que ele aproveite ao máximo seu tempo de treinamento. Os drones também podem ser usados para rastrear o progresso de um atleta ao longo do tempo e fornecer *feedback* sobre seu

desempenho. Esse acompanhamento de trajetória automática do drone abre muitas possibilidades de utilização, facilitando o trabalho do controlador na coleta de dados.

## 1.1 CONTEXTO

Os drones têm sido um recurso cada vez mais utilizado para o acompanhamento de vários processos, através deles é possível fazer filmagens e fotografias aéreas com precisão e agilidade, os drones com sua visão privilegiada, associados a novas abordagens de uso e grandes capacidades de processamento, podem fazer filmagens de forma simples para a recreação ou podem ser importantes ferramentas para pesquisa de dados, o drone com função *follow me* pode ser usado para fornecer informações sobre velocidade e distância percorrida. Essas informações podem ser utilizadas para melhorar o treinamento e a estratégia das equipes.

## 1.2 OBJETIVO GERAL

Este trabalho tem como objetivo construir um drone do tipo faça você mesmo (DIY, Do inglês *Do It Yourself*), e projetar um sistema acompanhamento entre o drone e um celular com sistema de posicionamento global (GPS). O projeto tem como finalidade enviar as coordenadas de posicionamento do celular recebidas pelo GPS, e gerar uma resposta automática do drone quando a posição do GPS em solo alterar, possibilitando o acompanhamento da trajetória.

## 1.3 OBJETIVOS ESPECÍFICOS

- a) Realização de pesquisa sobre drones DIY;
- b) Realizar pesquisa sobre GPS;
- c) Construir um drone DIY;
- d) Conectar o drone com um GPS ao celular.
- e) Realizar testes de integração entre o drone e o celular.

## 1.4 JUSTIFICATIVA

Atualmente, o mundo vive uma época de avanços tecnológicos rápidos sem precedentes. Vale a pena revisitar os velhos desafios à luz dos novos aplicativos, redes sociais, impressoras 3D, robots, Internet das Coisas (IoT) etc. Tentar identificar aí uma oportunidade disfarçada (EXAME, 2017). O projeto está intrinsecamente ligado aos princípios fundamentais da engenharia Elétrica/eletrônica, destacando-se nos campos de controle, comunicação e processamento de sinais. A implementação bem sucedida desse sistema abrirá novas possibilidades de uso para os drones, seja como ferramentas para uso profissional, seja como ferramentas recreativas onde pessoas podem usar para fazer vídeos de suas atividades.

## 1.5 ESTRUTURA DO TRABALHO

Este estudo é estruturado em cinco capítulos, abrangendo desde os conceitos teóricos até a aplicação e avaliação do protótipo criado.

A introdução corresponde ao primeiro capítulo, nela contém o objetivo geral e os objetivos específicos, basicamente traz as informações pertinentes do trabalho.

O segundo capítulo introduz a base teórica necessária para a compreensão do estudo, voltada para a história e dados importantes para o projeto.

Seguindo, o terceiro capítulo detalha o processo de elaboração do projeto, incluindo as hipóteses iniciais e a execução prática da construção.

No quarto capítulo, os ensaios realizados com o protótipo são descritos, com a apresentação dos dados obtidos através de tabelas e gráficos.

Concluindo, o quinto capítulo sintetiza as descobertas e considerações finais do estudo.

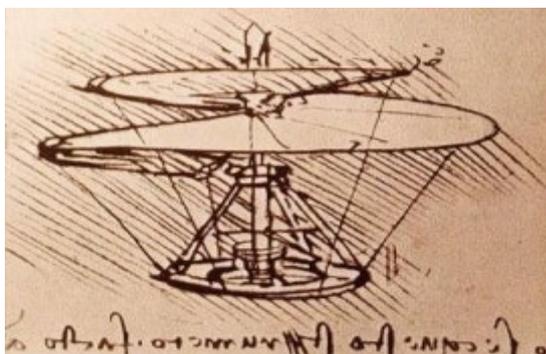
## 2 REVISÃO DA LITERATURA

Neste capítulo são apresentadas as disposições técnicas que são essenciais para compreender a construção do projeto. Explorar detalhes e aspectos técnicos que formam a base do trabalho, fornecendo insights sobre os componentes, tecnologias e princípios que orientam a criação e funcionamento dos VANTs, GPS, transição de dados e processamento dos referidos dados.

### 2.1 HISTÓRIA DO VANT

O homem sempre sonhou em alçar voo como os pássaros e houve muitas tentativas de criar uma forma que tornasse isso possível. Desde o projeto do Parafuso Aéreo de Leonardo da Vinci apresentado na Figura 1, até a construção do 14bis de Santos Dumont representado na Figura 2, que tomou isso realidade, ocorreram várias tentativas diferentes de construir uma máquina que pudesse conquistar os céus. Com a descoberta de meios para tal, impulsionou grandes avanços e muitas mudanças no mundo (EXAME, 2017).

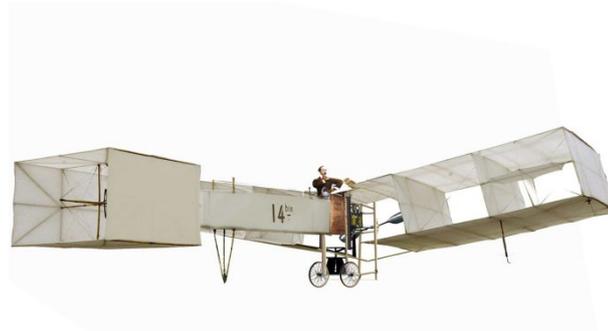
Figura 1 –“Parafuso aéreo” de Leonardo Da Vinci.



Fonte: Adaptado de Leishman (2023).

A história dos VANTs é tão rica quanto sua evolução tecnológica, porém há certa dificuldade de precisar com exatidão quando foram inventados. Estima-se que o primeiro tenha sido o “*Aerial Target*”, desenvolvido em 1917 para o treinamento dos soldados no uso de armas antiaéreas. Com um longo histórico de uso militar, os VANTs começaram a conquistar mercado com o início da fabricação recreativa. Atualmente contam com diversos nichos de aplicação e tem até mesmo modalidades de corridas. (PROPEL, 2023).

Figura 2– 14-BIS de Santo Dumont.

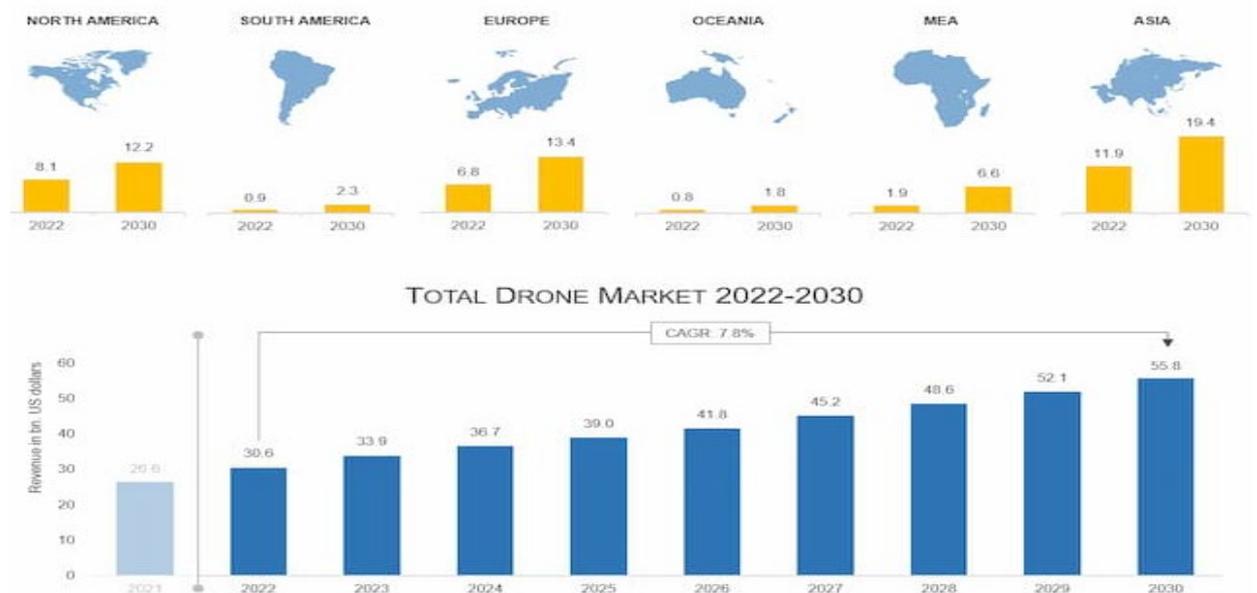


Fonte: Museu do Amanhã (2023).

O VANTs estão completamente inseridos nos mais variados setores da economia, apresentando novas abordagens e possibilidades de trabalho e reduzindo significativamente os custos na comparação com modelos tradicionais (GRANEMANN, 2023). O mercado de drones apresenta um crescimento nos últimos anos conforme apresentado pela revista DRONE INSIGHT (2022), o mercado global deva saltar de US\$ 30,6 bilhões em 2022 para US\$ 55,8 bilhões em 2030 seguindo uma taxa de crescimento anual de quase 8%.

O Brasil é hoje o principal mercado de drones da América do sul, com um faturamento anual estimado em US\$ 373 milhões, conforme apresentado na Figura 3.

Figura 3– Mercado Global de drone.



Fonte: Droneii (2022).

Segundo a revista ASAS (2023), o departamento de controle do espaço aéreo (DECEA) divulgou que em 2022 o Brasil teve 310 mil operações de VANTs homologados, um aumento de 27,8% frente a 2021. Entre 2017 a 2022, o crescimento chegou a 1.474%. Os dados são referentes à voos inseridos no sistema para solicitação de acesso ao espaço aéreo brasileiro por aeronaves não tripuladas. (REVISTA ASAS, 2023). No Gráfico 1 pode-se visualizar o crescimento.

Gráfico 1 - Solicitações de vôos no espaço aéreo no Brasil.



Fonte: Revista Asas, 2023.

### 2.1.1 VANT e drone

Neste subcapítulo, são apresentados detalhes sobre o conceito de VANT e drone, bem como suas principais características. Ao adquirir uma compreensão desses termos e das diferenças entre eles e as demais aeronaves, poderá então ser abordado a sua construção, o seu funcionamento e suas aplicações (GARRETT, 2015).

Para a distinção dos mesmos pode-se basicamente pensar da seguinte forma: se o seu uso for para recreação pode-se chamar de drone, porém, se sua finalidade for para outros fins como, por exemplo, comércio, pesquisa, experiências, entre outras, o mesmo passa a ser entendido como um VANT. Os VANTs de uso militar, por exemplo, podem ter outros componentes como armamento, ou seja, são capazes de levar uma carga útil, sua construção pode ser feita com motores a combustão, e podem percorrer grandes distâncias. (EBC, 2015).

Segundo GARRETT (2013) A associação mais simples para entender o que são drones, e mesmo para que servem, é lembrar de brinquedos de controle remoto. O conceito é

simples: com um controle via rádio, você pode manobrar um drone sem tocar nele. No geral, estes aparelhos são concebidos para realizar tarefas arriscadas ao ser humano ou ferramentas para trabalhos que ninguém quer realizar.

Conforme apresenta o portal EBC (2015), na tradução literal, a palavra drone significa zangão ou zumbido, e tem origem na língua inglesa, todavia nem mesmo é um termo técnico, drone é mais uma palavra comercial e está associado ao barulho que as hélices fazem para voar, produzindo um zumbido parecido com o som do inseto. A terminologia técnica para esses veículos seria *Unmanned Aerial Vehicle* (UAV), que traduzido seria Veículo Aéreo Não Tripulado (VANT), ou, alternativamente Veículo Aéreo remotamente Pilotado (VARP). Nos dias atuais já se tem um número significativo de drones incorporados à rotina das pessoas, e basicamente estes são brinquedos utilizados para filmagens e são manobrados com um controle via rádio.

### 2.1.2 Estrutura

Os drones podem ter variações em seus componentes dependendo da função para a qual são construídos, e isso muda o seu formato, tem uma variação da quantidade de motores podendo ter três motores (tricóptero), quatro motores (quadricóptero), seis motores (hexacóptero), ou oito motores (octacóptero). Os itens indispensáveis são os responsáveis por fazer o drone levantar vôo e deslocar-se, no Quadro 1 são apresentados alguns componentes (MARTINS, 2022).

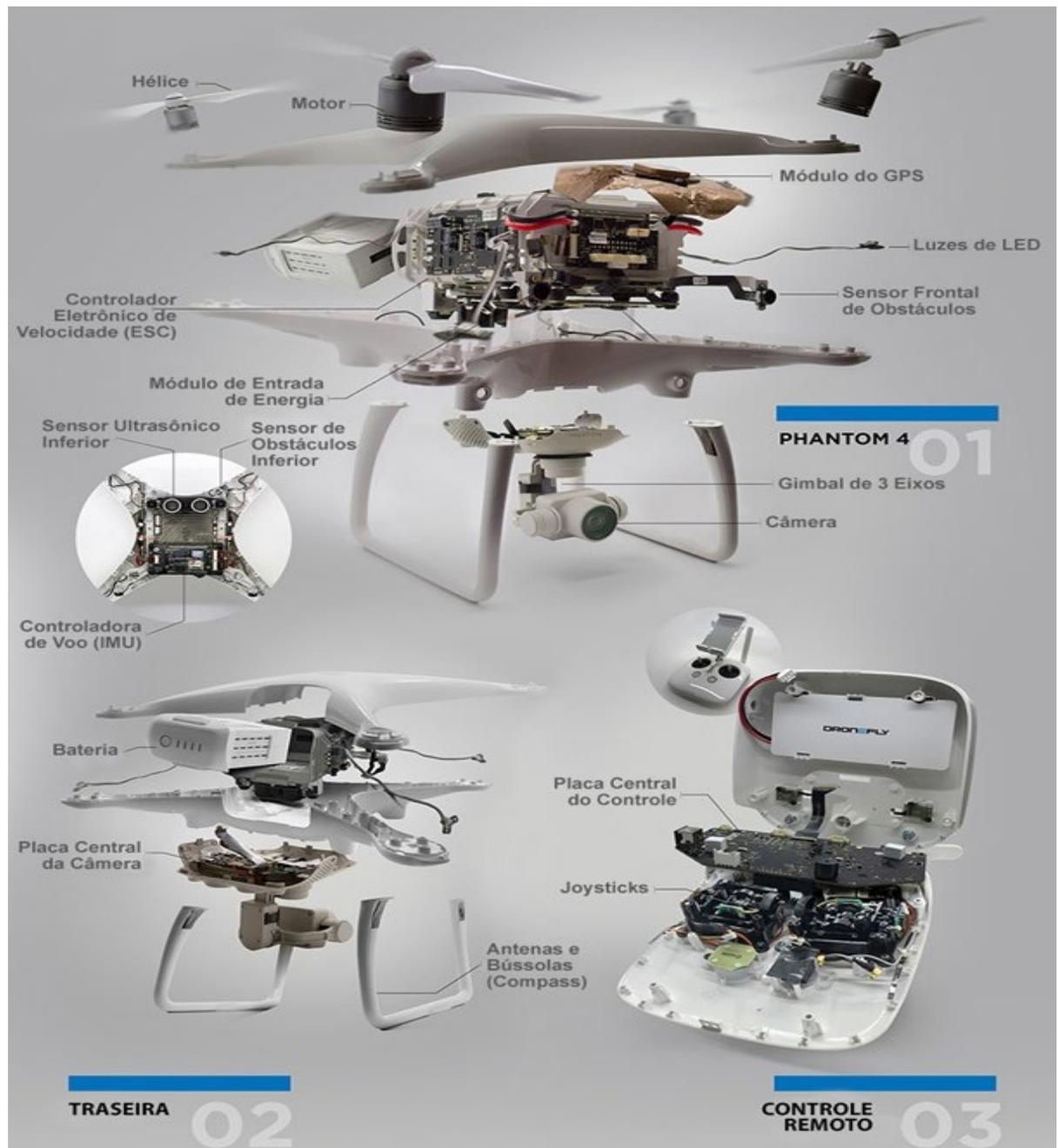
Quadro 1 – Componentes:

<b>Item</b>	<b>Tipo</b>	<b>Função</b>
Motor	Obrigatório	Item essencial para fazer o drone voar.
Bateria	Obrigatório	Item essencial para alimentar todo o drone.
Hélices	Obrigatório	Item essencial para o drone poder sair do chão.
Frame	Obrigatório	Item essencial para fixar as demais peças.
Controladora	Obrigatório	Item essencial para controle do drone.
Câmera	Opcional	Item opcional para obtenção de informações.
Rádio Controle	Opcional	Item opcional para o envio dos comandos.
GPS	Opcional	Item opcional serve para de opções de posição.
Sen. Anticolisão	Opcional	Item opcional que serve para evitar acidentes.

Fonte: Com dados adaptados de RUSSO (2021).

Na Figura 4 é apresentado um Drone comercial como exemplo, com uma vista explodida de sua construção para facilitar a visualização das partes. O modelo de drone é um DJI Phantom4, amplamente comercializado no Brasil, que pertence à categoria de quadricópteros (FlyPro, 2023).

Figura 4 – Componentes de um Drone.



Fonte: Adaptado de FlyPro (2023).

É importante destacar que além dos modelos comerciais, também existem diversas opções para aqueles que optam por montar seus próprios drones DIY que será apresentado na Figura 5. Esses drones DIY podem apresentar variações significativas em termos de características, construção e componentes uma vez que o responsável pelo projeto avalia quais partes serão incorporadas (Eletrogate, 2022).

Figura 5 – Drone DIY.



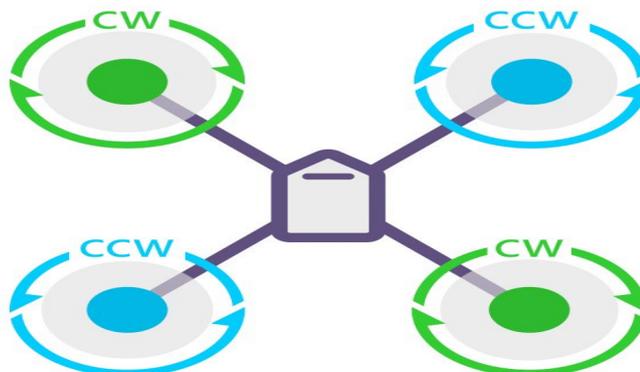
Fonte: kickstarter (2018).

### 2.1.3 Funcionamento

Para entender um pouco funcionamento do drone MARTINS (2022) trás alguns dados, sua sustentação no ar é gerada pelo movimento rotativo das hélices, um arranjo específico de funcionamento é necessário para evitar que ele gire no ar no mesmo sentido das hélices.

Um quadricoptero, por exemplo, tem-se um funcionamento de par, em formato de X, duas hélices giram em sentido horário e as outras duas em sentido anti-horário, conforme apresentado na Figura 6, esse sistema faz a compensação para que o drone permaneça estável no ar (MARTINS, 2022).

Figura 6 – Rotação das hélices.



Fonte: Blog Eletrogate (2022).

Para controlar o drone é preciso de uma controladora de voo, segundo o BLOG ELETROGATE (2022), é o cérebro do drone, a controladora é que faz a estabilização e compensações necessárias conforme instruções vindas de sensores contidos na própria placa, como acelerômetro, barômetro, giroscópio e magnetômetro, do sistema de GPS e telemetria, bem como do piloto através do rádio controle. Existem diversas controladoras no mercado: desde controladoras de entrada, como a KK2 que possui um microcontrolador Atmega de 8 bits e apenas mantém o drone estabilizado, até controladoras de luxo como a Pixhawk, que possui um microcontrolador ARM de 32 bits, além de diversas conexões de expansão para sensores. As controladoras são ilustradas na Figura 7. Controladoras mais avançadas permitem a instalação de sensores de obstáculos, conexão GPS, sonar e a seleção entre diversos modos de voo pré-definidos, adequando-se à necessidade do piloto. Esses modos de voo permitem que o piloto alterne entre um ou outro sensor, faça vôos auxiliados pelo GPS, missões automáticas e até mesmo o retorno e pouso da aeronave exatamente no local de decolagem.

Figura 7 – Controladoras Pixhawk e KK2.



Fonte: Blog Eletrogate (2022).

Conforme apresentado por MARTINS (2022) para transmitir e receber os dados precisa-se de um sistema de telemetria, que é um sistema baseado em transceptores, dispositivos capazes de transmitir e receber os dados vitais de navegação, sinal, bateria e diversas outras informações. Essas informações podem estar vinculadas somente com o sistema de rádio controle, limitando-se à intensidade do sinal RSSI (*Received Signal Strength Indication*) chegando até controladora do drone, o que permite a alteração de parâmetros em voo, configuração e cancelamento de missões automáticas e muito mais.

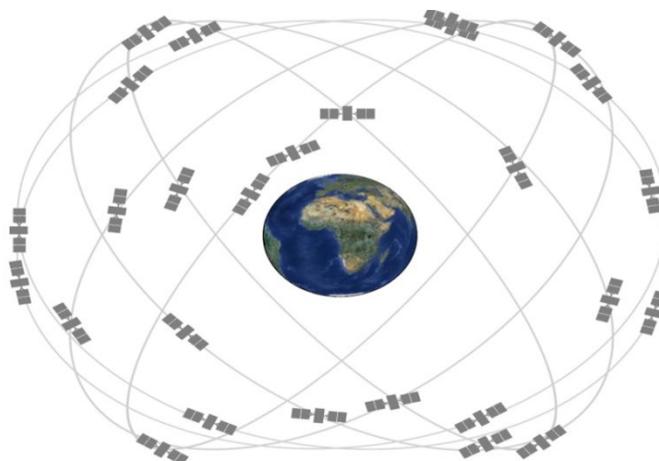
MARTINS (2022), explica que o GPS e Bússola (Compass) são os sistemas de posicionamento do drone. Através deles a aeronave consegue identificar onde está no globo e o norte magnético. Com um sistema de GPS embarcado, os drones são capazes de manter sua posição travada em voo, executar missões automáticas, onde o piloto define a rota de voo desejada e o drone segue rigorosamente os locais marcados. Outra função muito interessante que o GPS permite é o retorno automático em caso de perda de sinal de rádio ou bateria próxima do fim.

## 2.2 POSICIONAMENTO

Para determinar a posição instantânea precisa-se de um GPS, segundo GUITARRARA (2023), GPS é um sistema de localização que se utiliza de uma rede de 24 satélites artificiais, que são exemplificados na Figura 8, para captar dados e fornecer informações a respeito da posição de qualquer ponto na superfície terrestre por meio da latitude e da longitude, isto é, das coordenadas geográficas. Além disso, o GPS é capaz de oferecer, com precisão, a

altitude e o horário. Esse sistema funciona ininterruptamente e todos os dados são informados ao usuário em tempo real.

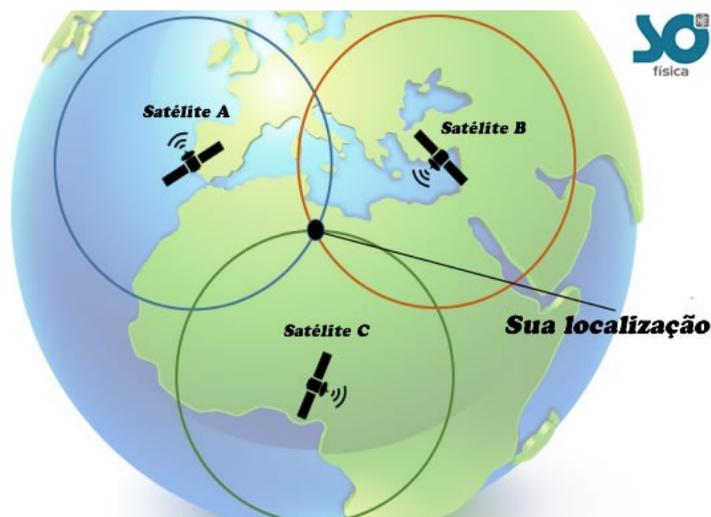
Figura 8 – Planos orbitais dos satélites de GPS.



Fonte: U.S. space force (2023).

GPS é um sistema de navegação por satélite introduzido pelos Estados Unidos em 1978. A tecnologia foi desenvolvida inicialmente para fins militares, mas se tornou amplamente utilizada em aplicações civis (TECNOBLOG, 2023). Para o fornecimento de dados é necessário o trabalho de quatro satélites em conjunto com as estações de controle, alocadas na superfície terrestre, e os receptores. São os receptores que interpretam as informações e realizam as conversões em coordenadas geográficas precisas (GUITARRARA, 2023). O funcionamento é apresentado na Figura 9.

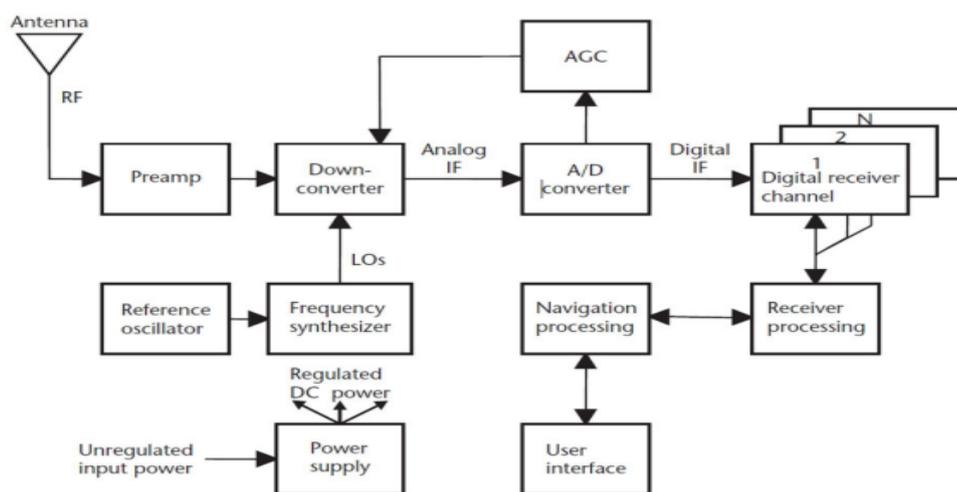
Figura 9 – Triangulação a partir dos satélites é à base do sistema do GPS.



Fonte: Só Física (2023).

No processo de triangulação, três satélites enviam o sinal para o receptor, que calcula quanto tempo cada sinal demorou a chegar até ele. Além da sua localização terrestre, o receptor GPS, também consegue saber a altura em relação ao nível do mar, porém para isso é necessário um quarto satélite (SÓ FÍSICA, 2023). A Figura 10 apresenta um diagrama de blocos genérico do receptor.

Figura 10 – Receptor GPS Digital Genérico Diagrama de Blocos.



Fonte: KAPLAN, HEGARTY pg. 139(2017).

Tanto os satélites como os receptores GPS possuem um relógio interno que marca as horas com uma enorme precisão, em nanossegundos. Quando o Satélite emite o sinal para o receptor, o horário em que ele saiu do satélite também é enviado, ao captar os sinais dos satélites, o receptor calcula a distância entre eles pelo intervalo de tempo entre o instante local e o instante em que os sinais foram enviados. Levando em conta a velocidade de

propagação do sinal, o receptor pode situar-se na intersecção desses dados, permitindo identificar exatamente onde o aparelho se encontra na Terra para que a posição do receptor seja sempre atualizada, os envios desses sinais ocorrem constantemente em uma velocidade de 300 mil quilômetros por segundo (velocidade da luz) no vácuo (SÓ FÍSICA, 2023).

O protocolo utilizado segundo SIMONI, SCARAMELLA (2021), foi desenvolvido pela *National Marine Electronics Association* (NMEA), define ambos os requisitos elétricos de sinal e formato das mensagens para viabilizar a comunicação, com foco na configuração “um produtor, múltiplos consumidores”, entre dispositivos eletrônicos comumente empregados em aplicações marítimas, entre eles, o GPS. O estilo de formatação de mensagens NMEA acabou se popularizando e se tornando o principal padrão de interfaceamento com GPS comerciais para aplicações gerais. O padrão estipula o uso de mensagens com caracteres ASCII (Código Padrão Americano para o Intercâmbio de Informação) seguindo o formato “\$aacc,c-c\*hh<CR><LF>”, cujo significado, para cada bloco de caracteres, é apresentado no Quadro 2.

Quadro 2 – Formatação NMEA.

ASCII	Descrição
\$	Começo de uma sentença.
aacc	Campo de endereçamento. O “aa” identifica o produtor, enquanto o “cc” descreve a informação contida na mensagem.
,	Delimitador. Este caractere delimita o começo de um novo campo, com exceção dos campos de endereço e CRC.
c-c	Este campo representa a sequência de informação efetivamente transmitida. Consiste de uma série de dados, também em ASCII, separada por caracteres delimitadores. Estes dados não possuem identificadores e deverão ser interpretados por meio de sua ordem de acordo com o que é estabelecido para uma mensagem do tipo “cc”.
*	Delimitador de check sum. Este caractere representa que os dois caracteres subsequentes contêm o check sum do pacote.
hh	O check sum é calculado por meio de uma operação OR exclusivo entre todos os bytes do dado da mensagem, ou seja, entre os delimitadores \$ e *. Cada nibble do resultado é então convertido para um caractere ASCII (0-9, A-F (upper case)).

<CR><LF>	Os caracteres carriage return e line feed indicam o fim do pacote.
----------	--

Fonte: NMEA (2002).

Especificamente, para mensagens GPS os caracteres “aa” de identificação do produtor são “GP”. Para a aplicação de rastreamento de veículos, são necessários dados de posicionamento, estampa de tempo e velocidade (em km/h). Além disso, valores de erros de fixação presentes na mensagem GSA podem ser utilizados para checar a validade dos dados obtidos (SIMONI, SCARAMELLA, 2021).

As mensagens de localização geográfica, latitude e longitude são chamadas GLL, a seguir um exemplo:

\$GPGLL,4916.45,N,12311.12,W,225444,A,\*1D

No Quadro 3 é apresentado o significado da sentença:

Quadro 3 – Significado da sentença.

“\$” :	Início da sentença.
“GP” :	Fonte de sentença (receptor GPS).
“GLL” :	Identificação da sentença.
“4916.45” :	Latitude em graus (49) e minutos (16.45).
“N” :	Latitude referente ao hemisfério norte .
“12311.12” :	Longitude em graus (123) e minutos (11.12).
“W” :	Longitude situada a oeste de Greenwich.
“225444” :	Horário em que a posição foi obtida ( 22h54m44s - UTC).
“A” :	Indica validade da posição (A → válido, V → inválido).
“*” :	Indica o último campo.
“1D” :	Soma de verificação (Checksum).

Fonte: NMEA (2002)

### 2.3 COMUNICAÇÃO

Segundo CARVALHO (2008), Comunicação é o processo pela qual a informação gerada em uma fonte é transferida a outro ponto, o destino da informação. Desde os

primórdios da humanidade o ser humano buscou se comunicar com seus semelhantes. As informações geradas na mente de um ser humano (fonte) são manifestadas fisicamente através de uma mensagem, que pode assumir várias formas, tais como sons (voz ou música), imagens, textos e muitas outras. Quando a fonte e o destino estão próximos um do outro a comunicação é direta e imediata, quando a distância entre elas aumenta o processo de comunicação se torna difícil, se não impossível. Há então a necessidade de um sistema de telecomunicação, um conjunto de meios e dispositivos que permita fonte e destino se comunicar.

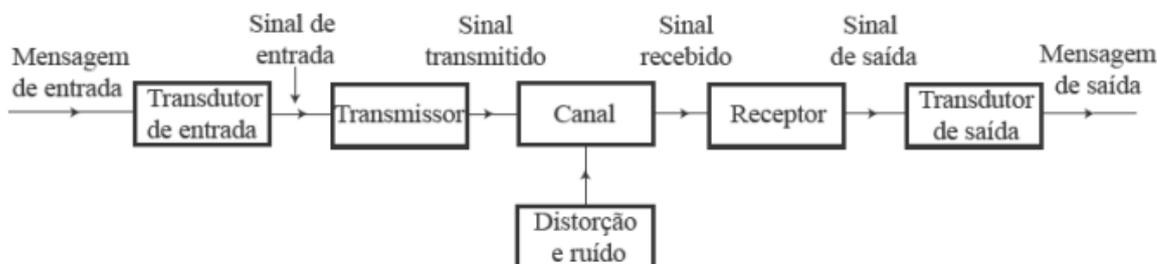
A comunicação sem fio tem sido um ponto importante na sociedade moderna desde a invenção do telégrafo. A telegrafia sem fio, também conhecida como radiotelegrafia, envolve a transmissão de ondas de rádio pelo ar em pulsos curtos e longos. Esses “pontos” e “traços” também conhecidos como Código Morse eram então captados por um receptor e traduzidos em texto por um operador de recepção. Esse método de comunicação permitiu que os seres humanos se comunicassem através de grandes distâncias com relativa facilidade (ESTES, 2018).

Com o passar dos anos houve muitos avanços, e em 1971, a Universidade do Havaí colocou em prática um sistema chamado ALOHAnet, que conseguia transmitir, sem fio, pequenos pacotes de dados entre computadores da instituição. Em 1989, o órgão dos Estados Unidos responsável pelas telecomunicações autorizou o uso de três faixas de frequência para o desenvolvimento de um padrão que se tornaria futuramente o Wi-Fi. No ano seguinte, o Institute of Electrical and Electronics Engineers (IEEE) criou um comitê que definiria esse padrão, criando aí o embrião do Wi-Fi (EXPLORERNET, 2022).

Segundo CARNEIRO (2017), a transmissão de dados entre drone e controle é chamada de telemetria. Os módulos de telemetria podem usar rádio, Bluetooth e até Wi-Fi Banda ISM (Industrial Scientific and Medical) para enviarem a uma estação de solo (Geralmente plugada a um computador) as informações sobre a situação do drone em tempo real. Este tipo de módulo trabalha em conjunto com o controlador de voo, recebendo o máximo de informações possíveis e as repassando ao controle de solo. A telemetria pode dar informações vitais como, por exemplo, estado atual das baterias e tempo restante estimado de voo. Além de informações sobre o estado do drone, alguns módulos especiais de telemetria possuem capacidade de transmitir vídeos em tempo real. Este tipo de módulo pode auxiliar em um resgate aéreo (permitindo o reconhecimento de regiões e pessoas em perigo, por exemplo) assim permite que o piloto tenha uma visão como se estivesse

embarcado na aeronave Visão em primeira pessoa (FPV). Na Figura 11 é apresentado um diagrama genérico de sistemas de comunicação.

Figura 11 – Diagrama Sistemas de comunicação.



Fonte: Adaptado de Lathi, pg 22 (2012).

Segundo LATHI (2012), um sistema de comunicação é composto por um transmissor, um canal, um receptor e o destino, o transmissor modifica o sinal em banda base para transmissão eficiente. O transmissor pode consistir em um ou mais subsistemas: conversor A/D, codificador e modulador. De forma semelhante, o receptor pode ser um demodulador, um decodificador em um conversor D/A. O canal é meio de escolha no qual os sinais elétricos na saída do transmissor são transportados. O receptor processa o sinal recebido do canal, revertendo às modificações feitas pelo transmissor no sinal e removendo as distorções feitas pelo canal. A saída do receptor é alimentada ao transdutor de saída, que converte o sinal elétrico a sua forma original, a mensagem, E por último temos o destino que é a unidade a qual a mensagem é endereçada.

Para realizar a transmissão de dados ou telemetria existem vários tipos de redes sem fio na banda ISM, como Bluetooth, Wi-Fi, ZigBee, SigFox, LoRa entre outras (MORGADO, 2010).

### 2.3.1 Bluetooth

Segundo MARTINS (2022) Bluetooth é um padrão para comunicações sem fio baseadas em um sistema de rádio projetado para conectividade de curto alcance para dispositivos pessoais portáteis. Ele define toda uma pilha de comunicação que permite que os dispositivos se encontrem e anunciem os serviços que oferecem. O Bluetooth é amplamente usado em tecnologia WPAN (rede de área pessoal sem fio, do inglês - Wireless Personal Area Networks), também conhecida como rede de curta distância sem fio. O

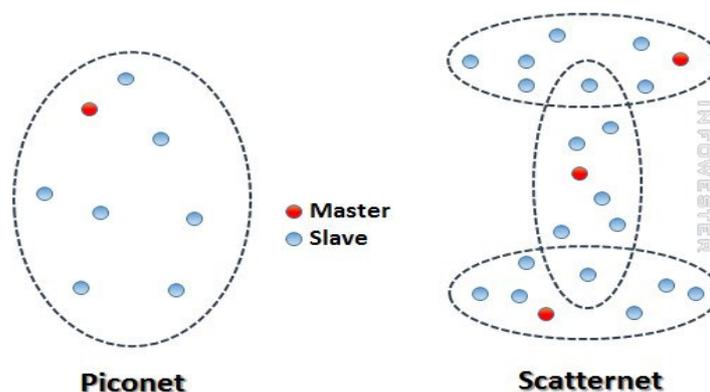
padrão IEEE 802.15.1 especifica a operação e a arquitetura dos dispositivos Bluetooth, mas a operação se refere apenas à camada física e à camada de controle de acesso ao meio (MAC – Do inglês Media Access Control). As camadas de protocolo e aplicativos são padronizadas pelo Bluetooth SIG (Special Interest Group). Os canais são acessados usando uma técnica espectro de difusão em frequência variável com uma taxa de sinal de 1 Mb/s, usando a modulação Gaussian shaped frequency shift keying (GFSK).

Conforme apresentado por NISHIKIORI (2022), o Bluetooth é uma tecnologia de comunicação sem fio presente em diversos produtos atualmente no mercado, de smartphones a caixas de som, fones de ouvido, passando por pulseiras fitness e outros dispositivos com internet das coisas (IOT). Desde seu lançamento, ele é gerido por uma organização própria, a Bluetooth SIG, que tem empresas como Microsoft, Apple e Intel como membros.

Cada dispositivo que usa Bluetooth possui um pequeno microchip que pode enviar dados e sinais de voz. Um dispositivo funciona como mestre em qualquer configuração típica e um ou vários dispositivos funcionam como escravos (EMBARCADOS, 2021).

Conforme apresenta a página EMBARCADOS (2021) este dispositivo mestre utiliza um software gerenciador de link para diferenciar outros dispositivos Bluetooth e criar links para receber e enviar dados. Os sistemas Bluetooth incluem pilhas de protocolo, transceptores e bandas base e podem criar uma rede compacta com alguns dispositivos. Os sistemas criam uma grande rede distribuída que consiste em numerosas redes Pico independentes e um cluster de piconets interconectadas denominado scatternet. Um sistema Bluetooth simples inclui antenas, software, controle de link e gerenciamento de link. Na Figura 12 são apresentadas as redes piconets e scatternet.

Figura 12 –Piconet e Scatternet.



Fonte: Infowester (2023).

No Bluetooth, o transporte é feito por um dos 79 canais da banda 2,4 GHz ISM, a mesma usada por micro-ondas, telefones sem fio e walkie-talkies. Para evitar interferências, os canais podem ser trocados aleatoriamente a uma velocidade de até 1.600 saltos por segundo (NISHIKIORI, 2022). A banda de 2,4 GHz, na faixa de frequência ISM, é reservada para aplicações Industriais, Científicas e Médicas. Como ela é aberta e gratuita, garante o funcionamento do Bluetooth em escala global. O alcance dos seus sinais de rádio permite o estabelecimento de comunicações entre dispositivos até uma distância máxima de cerca de 100 metros (MARQUES, ALECRIM, 2023).

A tecnologia Bluetooth foi atualizada com o passar do tempo e surgiram novas versões que variam em aspectos como tecnologia de transmissão de dados e taxa de transferência. Elas são definidas como Bluetooth 1.0, 2.0, 3.0, 4.0, e 5.0 (MARQUES, ALECRIM, 2023).

Segundo a página da INTEL (2023), que é uma das criadoras do Bluetooth, atualmente, a tecnologia Bluetooth é oferecida em duas versões distintas, Bluetooth Classic e Bluetooth Low Energy (LE). O Bluetooth Low Energy é otimizado para comunicações intermitentes que preservam a duração da bateria, enquanto o Bluetooth Classic é usado para aplicações que exigem transferência de dados mais frequente e persistente. O Bluetooth Classic tem até 79 canais de 2,4 GHz à sua disposição, enquanto o Bluetooth Low Energy tem até 40 canais para alternar.

As duas versões mais usadas são Basic Rate ou Enhanced Data Rate (BR/EDR), e Low Energy (BLE), o Bluetooth BR/EDR estabelece uma conexão sem fio contínua e de curto alcance. A taxa de dados EDR de 2-3 Mbits torna-o ideal para casos de uso como streaming de áudio. BLE permite conexão de rádio de longo alcance em rajadas curtas, tornando-o ideal para aplicações de Internet das Coisas (IoT). Os beacons Bluetooth são utilizados para aplicações baseadas em localização interna, detecção de atividade e detecção de proximidade. O uso de BLE está se tornando muito popular à medida que diferentes indústrias estão adotando para aplicação, como solução de rastreamento de ativos, unidades de manufatura, sistema de posicionamento de triangulação tridimensional usando beacons baseados em BLE (EMBARCADOS, 2021). Na Figura 13 é apresentada a foto de um módulo Bluetooth Low Energy.

Figura 13 – Módulo Bluetooth Low Energy - HC-08.



Fonte: Robocore (2023).

### 2.3.2 Wi-Fi

Segundo a INTEL (2023) o Wi-Fi é usado para conexões de internet e rede entre muitos dispositivos, enquanto a tecnologia Bluetooth é usada para uma variedade de casos de uso de dispositivo para dispositivo. O Wi-Fi opera nas faixas de frequência de 2,4 GHz, 5 GHz e 6 GHz, enquanto a tecnologia Bluetooth opera apenas na faixa de frequência de 2,4 GHz. A tecnologia Bluetooth destina-se a aplicações de baixa latência que lidam com quantidades menores de dados, enquanto o Wi-Fi fornece largura de banda e capacidade para mover arquivos grandes com mais rapidez e facilidade.

A maioria dos dispositivos eletrônicos hoje em dia tem interfaces Wi-Fi integradas, como computadores pessoais, consoles de videogame e smartphones, etc. Eles se conectam a recursos de rede através de um ponto de acesso à rede sem fio. Esses pontos de acesso têm um alcance de aproximadamente 20 metros em ambientes internos e um alcance maior em ambientes externos, geralmente de 100m a 300m (EMBARCADOS, 2021). No Quadro 4 é são apresentados os principais protocolos de Wi-Fi e suas características.

Quadro 4 - Principais Protocolos Wi-Fi.

<i>Geração/Padrão IEEE</i>	<i>Frequência</i>	<i>Taxa máxima de link</i>	<i>Ano</i>
Wi-Fi 4 (802.11n)	2,4/5 GHz	72–600 Mbit/s	2009
Wi-Fi 5 (802.11ac)	5 GHz	433–6933 Mbit/s	2014
Wi-Fi 6 (802.11ax)	2,4/5 GHz	600–9608 Mbit/s	2019

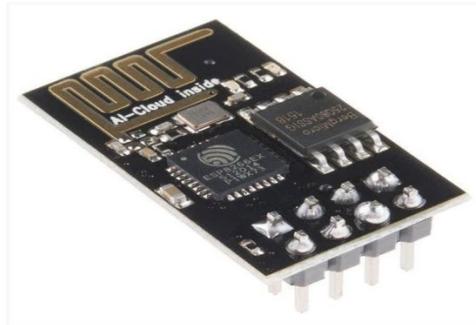
Fonte: Intel (2023).

Os dispositivos Wi-Fi operam em um canal fixo, em vez de alternar entre os canais com frequência, como os dispositivos Bluetooth, todas as redes Wi-Fi são sistemas Time

Division Duplex (TDD) centrados em contenção, onde estações móveis e pontos de acesso competem para usar o mesmo canal (INTEL, 2023).

O Wi-Fi também permite a comunicação direta de um computador (ou dispositivo) para outro sem ponto de acesso intermediário. Isso é chamado transmissão Wi-Fi Direct (INTEL, 2023). Na Figura 14 é apresentado um módulo Wi-Fi.

Figura 14 – Módulo Wi-Fi serial ESP8266.



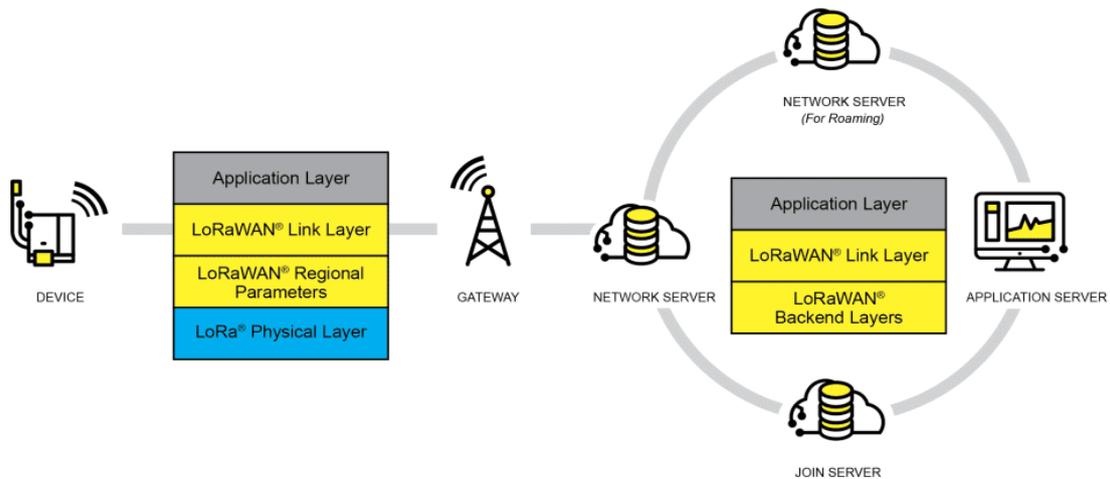
Fonte: Eletrogate (2023).

### 2.3.3 LoRa

A rede *Long Range* (LoRa) é uma solução sem fio sub-GHz em frequência não licenciada que endereça demandas para conexão entre dispositivos para aplicações de baixo consumo, longa distância e baixo custo de infraestrutura (CENTENARO et al., 2015).

A LoRa usa espectro de rádio não licenciado nas bandas ISM para permitir baixa potência, comunicação de área ampla entre sensores remotos e gateways conectados à rede. Ele usa tecnologia de espalhamento espectral. Sua frequência modulada utiliza ganho de codificação para aumentar a sensibilidade do receptor (EMBARCADOS, 2021). Na Figura 15 é apresentado a arquitetura da LoRa.

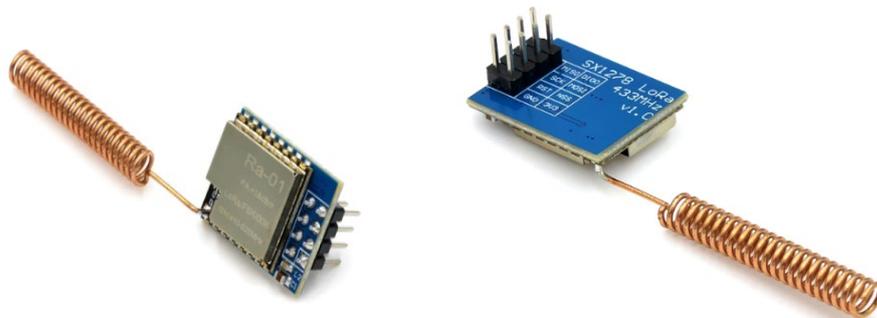
Figura 15 - Arquitetura de rede LoRaWAN.



Fonte: LoRa Alliance(2023).

LoRa é uma especificação de protocolo com código aberto construída pela LoRa Alliance que permite que outras empresas criem suas próprias redes IoT com base em suas especificações de tecnologia. Essa abordagem baseada em padrões para construir as ligações permite a configuração rápida de redes IoT públicas ou privadas em qualquer lugar usando hardware e software seguro, fornece localização precisa e funciona da maneira que você espera (LORA ALLIANCE, 2023). Os módulos LoRa são apresentados na Figura 16.

Figura 16 – Módulo LoRa.

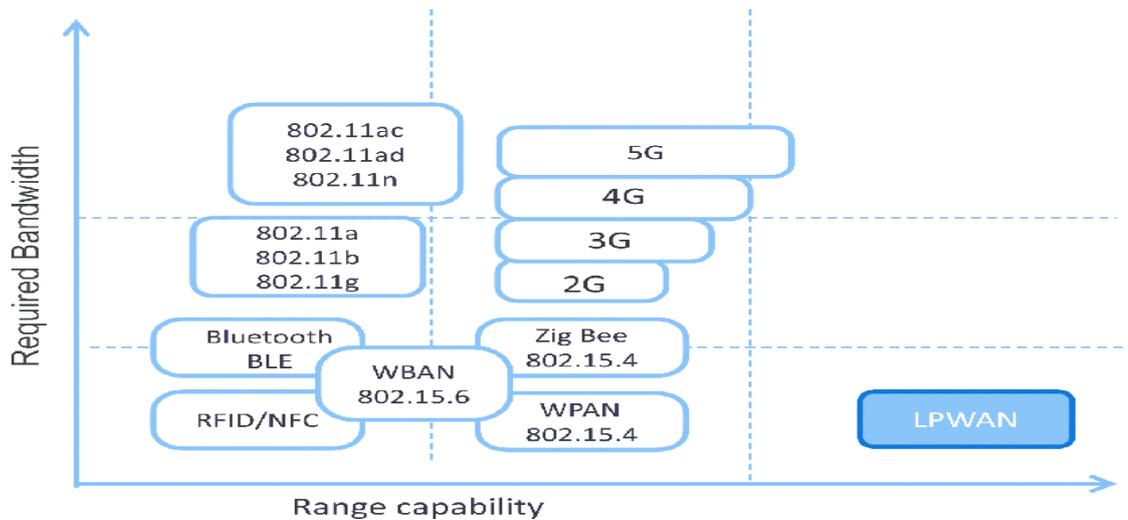


Fonte: AliExpress (2023).

Em alguns casos, as antenas LoRa podem ser combinadas com antenas celulares, pois as frequências podem estar próximas e a combinação de antenas fornecerá vantagens de

custo. As principais características do LoRa é que ele cobre um longo alcance de 15 – 20 quilômetros, pode se conectar a milhões de nós e sua bateria dura mais de 10 anos. (EMBARCADOS, 2021). Na Figura 17 é apresentado um comparativo entre o alcance e a largura de banda, e na Tabela 1 é apresentada uma visão geral.

Figura 17 - Alcance e largura de banda.



Fonte: UFRJ (2018).

Tabela 1: Comparação entre as redes sem fio

Protocolo	Alcance	Frequência	Taxa	Topologia
Wi-Fi	50 m	2,4/5 Ghz	1.300 Mbps	Estrela
BLE	80 m	2,4 Ghz	1 Mbps	Estrela/Malha
LoRaWAN	2-40 km	915 MHz (Brasil)	0,3-50 kbps	Estrela

Fonte: Pastório et al. 8 nov. 2023, p. 04.

## 2.4 DADOS

Segundo a página PANORAMA POSITIVO (2019) o processamento de dados é uma tarefa conjunta do hardware, como exemplo, computador, smartphone, tablet, notebook ou servidor, com o software que é o sistema informatizado ou aplicativo que fará a interface dos dados. O processamento de dados ocorre em diversas etapas. A entrada dos dados

acontece por meio do hardware que os coleta em sua memória e distribui para o processador e posteriormente para o software encarregado de fazer a sua recepção, análise, tratamento e dar saída em informações que vão gerar respostas ao sistema.

Será utilizado neste trabalho processamento em tempo real, que é o processamento imediato e sequenciado de registros. Ao concluir o processamento de uma informação, é disparado um mecanismo de processamento de um novo registro. Um bom exemplo é o GPS, que vai processando informações do trajeto ao longo da jornada do veículo do usuário e liberando as orientações necessárias de direcionamento.

### 2.4.1 ESP32

Segundo MURTA (2024) a ESP32 foi lançada em 2016 pela empresa Chinesa Espressif, como sucessor do ESP8266, ela conta com dois processadores Xtensa de 32-bit LX6 trabalhando com a velocidade de processamento de até 240 MHz, mas sendo mais comuns trabalharem na velocidade de 160 MHz, a empresa lançou placas do tipo DEV que se popularizaram pela facilidade de conexão sem fio. Na Figura 18 é apresentado o chip ESP32.

Figura 18 – ESP32.

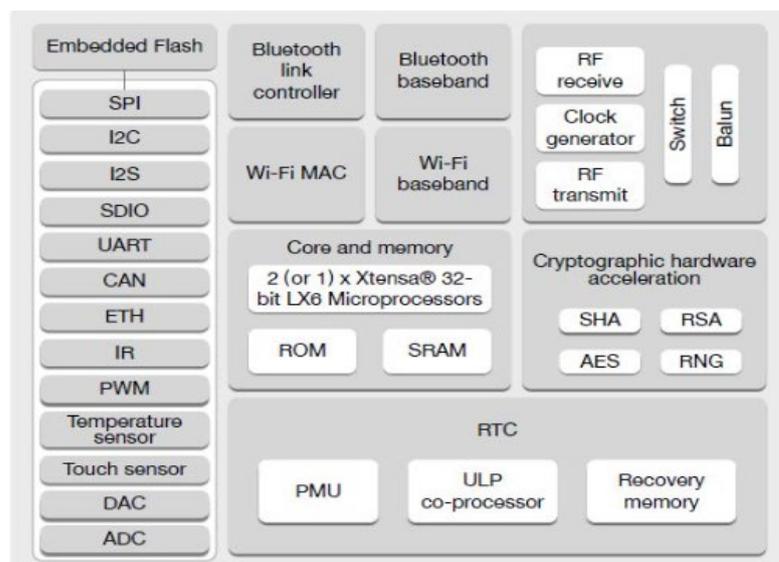


Fonte: Eletrogate (2018).

Segundo a ROBOCORE (2023) é possível criar projetos que geram notificações se seu dispositivo estiver em movimento, detectar quedas, alimentar banco de dados e dashboards a partir das leituras feitas, monitorar proximidade através do Bluetooth, entre outras

possibilidades. A vantagem na placa de desenvolvimento baseada no ESP32 é que existem bibliotecas prontas como TinyGPS ou a GPSneo, essas bibliotecas já fazem a manipulação das strings de dados do GPS. Na Figura 19 é apresentado diagrama de blocos do chip ESP32.

Figura 19 – Diagrama de blocos ESP32.



Fonte: Eletrogate (2018).

### 3 DESENVOLVIMENTO DO PROJETO

Neste capítulo serão abordados os aspectos que dizem respeito diretamente ao planejamento e desenvolvimento do projeto, nele serão apresentado os componentes, bem como a montagem e o desenvolvimento do projeto. Deve ser levado em conta que além da montagem o software de controle é muito importante para o projeto, pois tem um papel importantíssimo e precisa de significativos ajustes e configurações para o sucesso do projeto.

O controle da estabilidade de voo é muito importante e o software deve ser capaz de receber feedbacks do drone e conter parâmetros ajustáveis para corrigir os erros e fazer alterações, levando em conta que usaremos um software que é desenvolvido para uma grande gama de drones. Ele precisa conter inúmeros parâmetros configuráveis para que possa ser ajustado segundo as características particulares de cada drone.

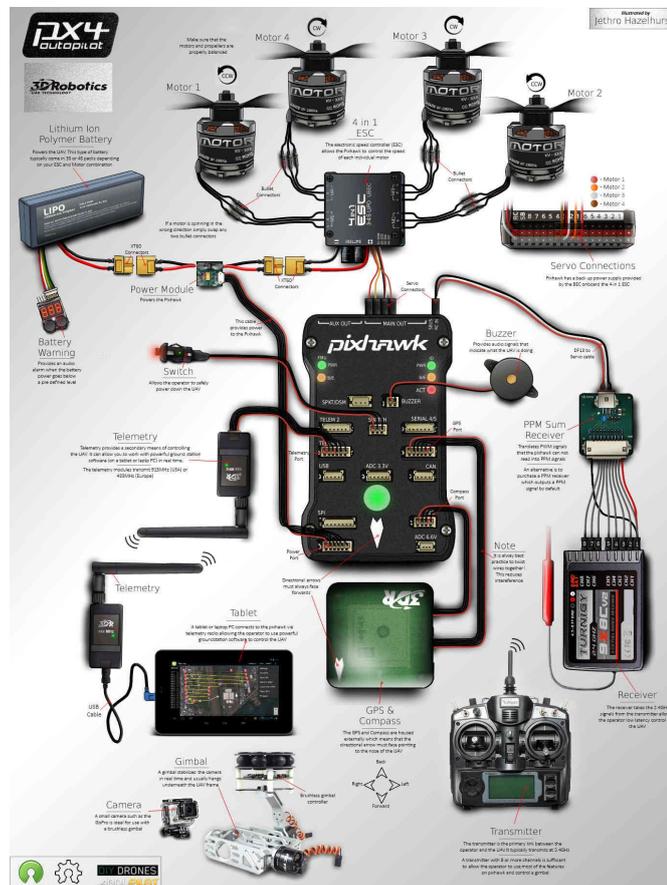
O hardware precisa ser construído com componentes de qualidade, pois precisa ter resistência, potência e sincronia de trabalho. A resistência do conjunto é importante, pois precisa ser considerado que até que os ajustes de voo sejam feitos o drone pode sofrer quedas, portanto precisa que a estrutura resista a essas possíveis quedas. A potência é

extremamente necessária neste projeto, visto que vencer a força da gravidade e manter-se em voo com uma estrutura que pode conter um considerável peso requer bastante força de empuxo. Sincronia dos componentes para manter o equilíbrio, altura e direção em voo é muito importante.

De nada adianta ter motores potentes se a bateria não tiver capacidade de corrente suficiente, logo teremos um gargalo na alimentação, que pode representar em baixa autonomia de voo e até mesmo acidentes com o superaquecimento da bateria pelo excesso de tempo trabalhando no seu limite. Da mesma forma as hélices devem ter capacidade de empuxo suficiente para o peso do conjunto, uma vez que se seu diâmetro se for muito pequeno irá requerer maior número de rotações dos motores, consumindo assim mais bateria e logo diminuindo a autonomia de voo.

Para o desenvolvimento deste projeto será usado uma controladora Pixhawk, por ser uma das controladoras com melhor custo benefício no mercado, além de ter uma boa gama de periféricos e sensores disponíveis. Na Figura 20 é apresentado um esquemático genérico dos componentes que podem ser utilizados com a Pixhawk.

Figura 20 – Esquemático genérico.

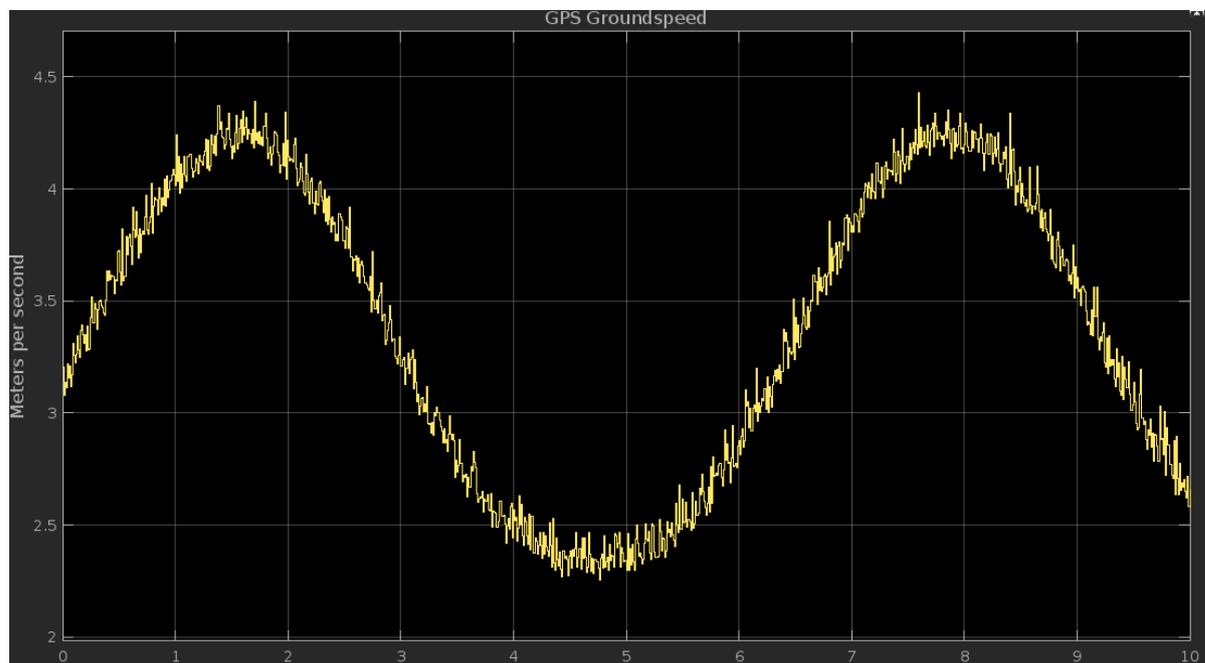


Fonte: ArduPilot (2024).

O projeto consiste em fazer a montagem de um drone do tipo DIY e após este estar totalmente operacional fazer com que ele seja capaz de fazer a comparação da posição instantânea com outro dispositivo baseado nos dados fornecidos pelo GPS, o primeiro desafio é que normalmente os dispositivos GPS tem uma margem de erro, e quando se pensa em comparar a duas posições essa margem de erro aumenta, outro grande desafio é o tempo de resposta do drone às alterações da estação de solo, se forem rápidas demais o drone pode acabar seguindo um ruído de comunicação ao invés da posição do outro dispositivo e vai necessitar muitas correções de posição, o que faria com que consumisse muita energia da bateria, se for lento demais vai acabar ficando desorientado.

No software Matlab foi feito uma simulação da leitura de um GPS, nesta simulação é usado um sinal senoidal e gerado uma perturbação na leitura, como poderemos notar na Figura a senoide tem diversas distorções, essa é a forma aproximada da leitura de um GPS. Na Figura 21 é apresentada a leitura de velocidade pelo GPS.

Figura 21 – Leitura de GPS.



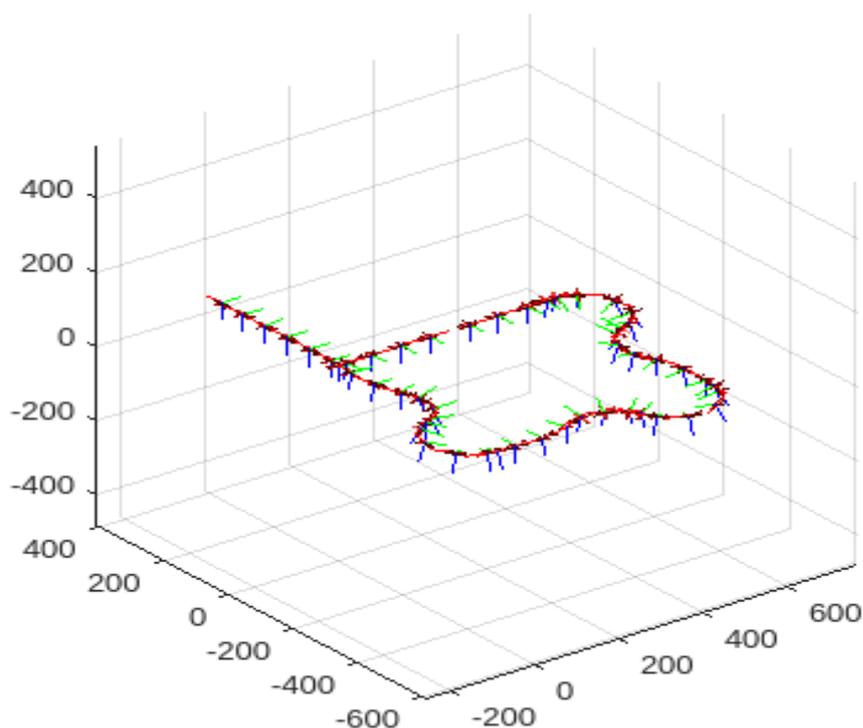
Fonte: Matworks (2024).

Pode-se notar pela figura que existe uma variação na leitura da posição durante a movimentação, essa seria a margem de erro do GPS, essa margem pode ser maior ou menor dependendo do tipo de receptor e se o GPS usa apenas um tipo de orientação, se ele

consegue usar mais de uma fonte de sinal ou se ele trabalha em conjunto com uma estação terrestre que faz a comparação da posição.

O drone possui o que é nomeado de “os seis graus de liberdade”, ou seja, ele pode realizar manobras em seis diferentes direções, sendo que ele pode se deslocar pelo eixo X,Y,Z e ainda pode fazer rotações por estes eixos, logo a trajetória do drone é bastante incerta e precisa realimentação dos sensores e de correções a cada instante para manter o voo estável e deslocar em uma determinada direção corretamente, na Figura 22 uma ideia de trajetória vetorial e da liberdade de manobra do drone. Na Figura 22 é ilustrada a Trajetória vetorial.

Figura 22 – Trajetoria vetorial.



Fonte: Matworks (2024).

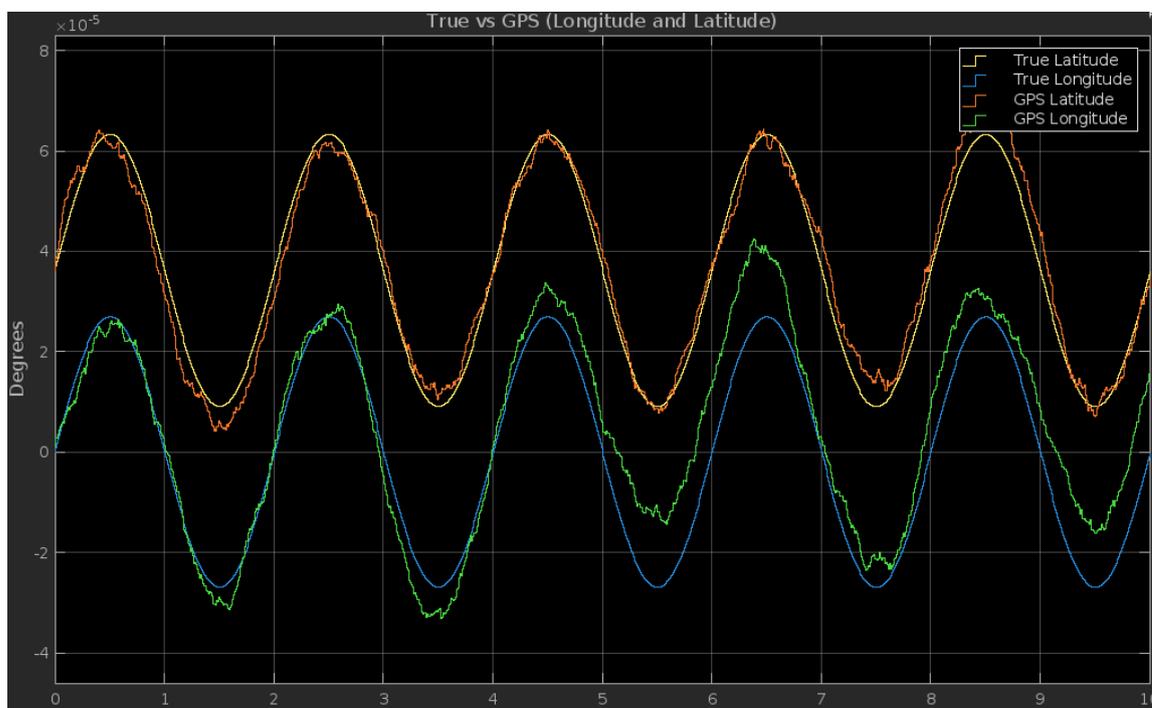
O acompanhamento precisa de constante realimentação da posição e mesmo assim haverá certo atraso no acompanhamento do drone em relação à posição do dispositivo acompanhado em solo, e esse atraso será maior ou menor dependendo do nível de movimentação e da velocidade do dispositivo em solo.

Não será feito nenhum tipo de sensor de colisão, logo é preciso garantir que o drone não colida com nada, uma vez que evitar obstáculos tornaria o projeto mais difícil, logo os testes são feitos em campo aberto, onde não existam postes, árvores, carros, casas, dentre outros

obstáculos que poderiam atrapalhar a simples função de seguir outro GPS, bem como áreas de sombra onde poderia ser perdida a conexão com os satélites que fazem o envio das mensagens NMEA.

Se o acompanhamento for feito em todos os eixos, será necessário controlar a altura do drone em relação ao solo, com isso haveria um aumento do número de variáveis. Logo mais complexo seria o sistema, se adicionar um sensor Anticolisão ao drone, precisaria aumentar esse sistema de controle para correções ainda maiores da trajetória, tornando o sistema mais complexo. Por hora não será feito esses controles, para reduzir a complexidade, e diminuir o tempo de atraso nas respostas. Foi realizada uma simulação da leitura de latitude e longitude do drone e é apresentado na Figura 23.

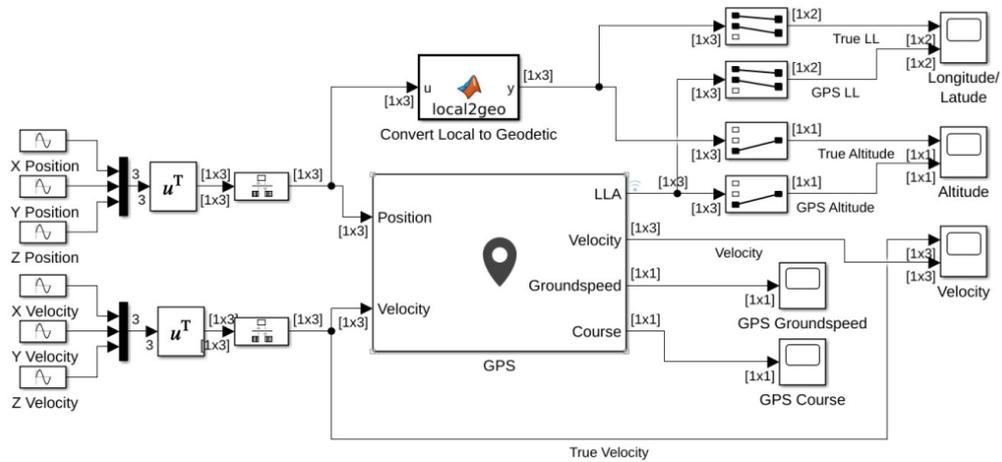
Figura 23 – Simulação de GPS.



Fonte: Matworks (2024).

A estrutura da simulação que foi utilizada para este trabalho é apresentada na Figura 24.

Figura 24 – Estrutura de simulação.



Fonte: MathWorks (2021).

Para começar o desenvolvimento deste projeto é preciso tratar as mensagens NMEA, para depois enviá-las ao drone como destino. Como um passo inicial foi utilizada uma placa de desenvolvimento ESP32 da marca LILYGO que conta com GPS e LoRa integrados, o GPS que vem nesta placa é o NEO-6M.

Utilizando a interface de desenvolvimento de códigos Arduino IDE foi testada a forma de fazer a leitura e tratamento das mensagens. Na Figura 25 é apresentada a placa LILYGO.

Figura 25 – Placa LILYGO.



Fonte: Proprio autor (2024).

Utilizando a interface do Arduino IDE e a utilização da biblioteca TinyGPS foi feito com sucesso o recebimento e tratamento das mensagens onde é exibido via serial monitor a posição de latitude e longitude. Na Figura 26 é apresentada a leitura das mensagens NMEA.

Figura 26 – Mensagens NMEA.

Sats	HDOP	Latitude (deg)	Longitude (deg)	Fix Age	Date	Time	Date Age	Alt (m)	Course ---	Speed from GPS	Card ----
****	6.1	-28.247917	-52.424221	141	06/29/2024	12:20:02	254	630.00	66.69	7.50	ENE
5	1.6	-28.247755	-52.424095	409	06/29/2024	12:20:06	529	630.00	39.70	11.78	NE
5	1.6	-28.247738	-52.424061	409	06/29/2024	12:20:07	529	630.00	41.85	11.30	NE
5	1.6	-28.247723	-52.424042	411	06/29/2024	12:20:08	530	630.00	43.14	10.45	NE
5	1.6	-28.247713	-52.424019	413	06/29/2024	12:20:09	531	629.90	46.53	9.89	NE
5	6.2	-28.247707	-52.424007	414	06/29/2024	12:20:10	532	630.00	49.01	8.09	NE

Fonte: Proprio autor (2024).

### 3.1 ESTRUTURA MECÂNICA/PROTÓTIPO

O drone DIY é composto pelos componentes:

- Frame: que é uma espécie de chassis.
- Motores: que são responsáveis por converter potencial elétrico em movimento mecânico.
- ESC: responsável por fazer o correto acionamento motores, bem como controlar a orientação e velocidade dos motores.
- Hélices: são responsáveis por transformar o giro dos motores em empuxo de ar.
- Placa controladora: é responsável por controlar todo sistema e receber os dados dos sensores e fazer o processamento.
- GPS: responsável pela localização.
- Módulo: de rádio comunicação, responsável pela comunicação com a estação terrestre.
- Controle remoto e receiver: são responsáveis pelo controle manual do drone caso o operador deseje controlá-lo ele mesmo.

Também é necessário um notebook ou um celular para a função de estação terrestre, as peças supracitadas serão abordadas a seguir, além destas especificadas ainda foram utilizados outros materiais como fios, parafusos, sintas, pilhas, conectores, etc.

### 3.2 HARDWARE

Conforme citado anteriormente foram necessários seguintes componentes: bateria específica para drone, motores brushless, controles eletrônicos de velocidade (ESC) para os motores, hélices, GPS, frame, controladora, controle remoto, módulos de transmissão de dados via radio frequência. A seguir os detalhes de cada um dos itens.

#### 3.2.1 Bateria

A bateria selecionada para o desenvolvimento do projeto é da marca Youme Power, ela é do tipo polímero de lítio (Lithium Polymer - LiPo), pois possui uma boa relação de peso e tamanho, com uma boa capacidade de energia. A bateria é dividida internamente em

4 células (4S), com uma alta capacidade de fornecer corrente elétrica, levando em conta que os motores do drone podem exigir uma grande demanda de corrente para fazer o drone voar. A tensão nominal da bateria é 14,8V, cada célula gera 3,7V ( $3,7 \times 4=14,8V$ ), com taxa de descarga de 5200 mAh, ou seja ela é capaz de alimentar um drone que consuma 5,2A por uma hora. Na Figura 27 é apresentada a bateria.

Figura 27 – Bateria.



Fonte: Próprio Autor (2024).

Na Figura 27 pode-se notar que existem dois conectores na bateria, um deles é para a carga/descarga, ou seja, é onde são conectados os periféricos e também onde é feito o carregamento da mesma. As fabricantes contam com diferentes tipos de conectores, este modelo conta com o padrão XT60, sendo este o de cor amarela, o outro conector de cor branca, é para equilibrar o carregamento da bateria, contando com cinco fios, um fio é comum a todas as células, e cada célula conta com um fio para controlar o carregamento, ou seja, medir se esta carregando de forma equilibrada todas as células. O peso da bateria é de 488 gramas, o tamanho é de 135x42x40 mm.

Para saber a capacidade de corrente da bateria é preciso fazer uma conta com os dados apresentados na própria bateria, os dados apresentado nela mostram que ela conta com uma capacidade de corrente drenada própria para drones é de 50C. A equação para o calculo segue o exemplo:

$$C_{max} = Tx * C \quad (1)$$

Onde  $C_{max}$  é corrente máxima, Tx é taxa de descarga e C a capacidade.

No caso da bateria selecionada, os dados na mesma apresentam a taxa de descarga de 50C e a capacidade de 5200 mAh, logo o cálculo fica da seguinte forma:

$$C_{max} = 50 (C) * 5200 (mAh)$$

$$C_{max} = 260 A$$

A capacidade da bateria é um importante fator para o drone, pois é baseando-se nela, e na potência dos motores que se dá a velocidade de rotação das hélices, então esses dois componentes precisam ter certo equilíbrio para que trabalhem em forma harmônica. Um exemplo de como funciona, supondo que os motores são 980 KV:

$$KV * V = RPM \quad (2)$$

$$980 KV * 14.8V = RPM$$

$$14504 RPM$$

Para o conjunto com essas especificações a rotação máxima das hélices será de 14504 RPM.

### 3.2.2 Controladora

A controladora selecionada foi a Pixhawk 2.4.8, por ser a mais dinâmica e completa comparada com as outras opções disponíveis no mercado, ela vem equipada com um processador de núcleo ARM Cortex M4 de 32 bits, 256 KB de RAM, 2 MB de memória flash, e com um coprocessador à prova de falhas de 32 bits. Na Figura 28 é apresentada a controladora.

Figura 28 – controladora Pixhawk.



Fonte: Próprio Autor (2024).

Esta controladora tem alguns sensores integrados na própria placa, estes sensores na maioria das vezes servem como sistema de redundância para diminuir falhas, como por exemplo, o acelerômetro que é um sensor integrado. Também há muitos sensores externos que podem ser adicionados ou não ao projeto, são sensores que precisam ser apenas conectados (plug and play), pois os drivers para o funcionamento já estão na memória da controladora. Entre os sensores estão barômetro, GPS, módulo para controle/gerenciamento da bateria, sensor ultrassônico, infravermelho, etc. O peso da controladora é 38 gramas e o tamanho é 50x15,5x81,5 mm.

Na placa encontram-se algumas interfaces, são elas, cinco entradas UART, sendo uma delas de alta potência e duas com controle de interrupção ou como descrito no manual do fabricante controle de fluxo por hardware. As entradas podem ser visualizadas na Figura 29.

Figura 29 – Entradas da controladora Pixhawk.



Fonte: Ardupilot adaptado (2024).

Também possui uma modulação por posição de pulso (PPM) de soma de sinal, onde todos os canais são transmitidos juntos, e cada canal é determinado pela largura desse pulso. Uma entrada indicadora de intensidade de sinal recebido (RSSI) que pode ser usada para modulação por largura de pulso (PWM) ou tensão para alimentação da placa. Dispõe uma entrada de circuito interno integrado (I2C), uma entrada serial de interface de periféricos (SPI), duas entradas do tipo *controller área network* (CAN) e uma entrada USB.

O sistema operacional da placa é gravado em um cartão SD. Ficam gravados no cartão os parâmetros predefinidos e as alterações feitas pelo usuário, que geralmente são feitas pela porta Micro-USB quando em solo, mas podem ser alteradas em voo com um sistema de comunicação que possa enviar informações a controladora. No projeto são usados módulos de RF para fazer essa comunicação.

A controladora contém LED's para que o usuário visualize o status do sistema e se tiver algum erro que precisa ser corrigido pode ser visualizado pelos status dos LED's como, por exemplo, o LED vermelho piscando indica erro nos parâmetros, o LED verde sólido indica sistema pronto. Os LED's são apresentados na Figura 30.

Figura 30 – LED's da controladora Pixhawk.



- 1 - Energia da Unidade de Gerenciamento de Voo (FMU).
- 2 - Modo Bootloader da FMU (piscando) ou Erro (contínuo).
- 3 - Energia da Unidade de Entrada/Saída (I/O).
- 4 - Modo Bootloader de I/O (piscando) ou Erro (contínuo).
- 5 - Atividade (piscando indica que todas as unidades estão respondendo).

Fonte: Ardupilot adaptado (2024).

### 3.2.3 GPS

O GPS utilizado no projeto é o NEO-M8N, ele já tem os drivers pré-instalados na controladora, sendo assim ele acaba sendo plug and play. Esse GPS obtém as coordenadas do GNSS, seu tempo de partida a frio, ou seja, quando precisa fazer a leitura dos satélites para detectar sua posição no globo leva em média 38 segundos. Nele contém uma bateria interna que o mantém funcionando por até seis dias, então dentro destes dias de duração da bateria o GPS tem uma partida a quente, que leva em média 1 segundo para enviar a posição para a placa controladora.

A taxa de atualização padrão é de 10 Hz, podendo ser alterado para até 18 Hz, o erro da posição fica abaixo de 2 metros, mas segundo o fabricante ele tem uma precisão da posição centimétrica. Esse módulo combina o posicionamento com o Real-Time Kinematic (RTK), o RTK é um conceito de veículo e base para uma melhor precisão da posição, nesse sistema ele compara a posição dele com a base fixa para comparar se está fazendo uma leitura correta dos dados recebidos dos satélites. Na Figura 31 é apresentado o NEO-M8N.

Figura 31– M8N.



Fonte: Próprio Autor (2024).

### 3.2.4 Radio controle

Embora o drone possa ser controlado sem o uso de um controle remoto, optou-se por adquirir um, tendo em vista uma maior segurança nos testes iniciais. O modelo selecionado foi o FS-I6 da marca Fly Sky (FS), sendo este um modelo bastante utilizado em aeromodelos, levando em conta seu custo benefício, este tem um custo médio de trezentos Reais, enquanto outros controles mais sofisticados podem custar mil Reais, com modelos podendo chegar até a faixa de cinco mil Reais. O controle vem com padrão de fábrica de seis canais, podendo por meio de atualização utilizar oito ou até dez canais de transmissão. O controle é apresentado na Figura 32.

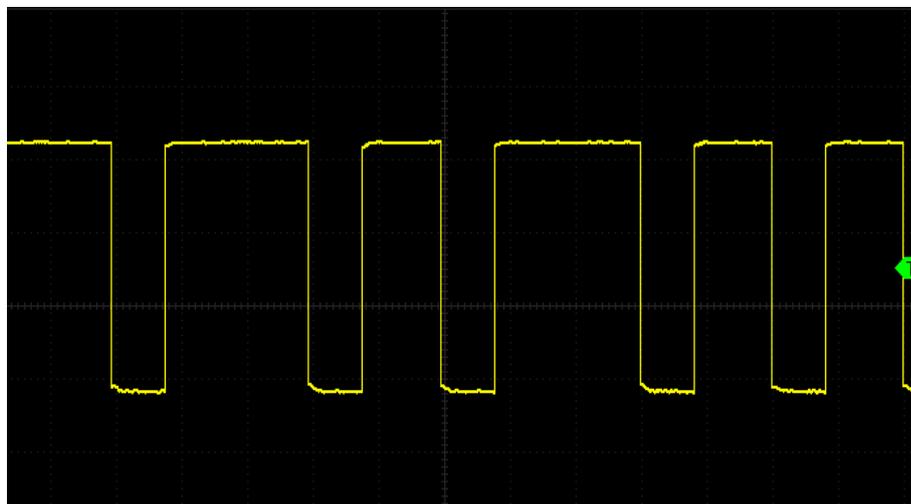
Figura 32 – Controle FS-I6.



Fonte: Próprio Autor (2024).

O controle trabalha na faixa de radio frequência (RF) de 2,4 GHz com uma largura de banda de 500 KHz, possui vários protocolos de transmissão, mas o que vai ser utilizado neste projeto é o PPM, por ser compatível com a controladora e também por ser um protocolo simples e confiável. Conforme explicado anteriormente a modulação PPM é a soma dos sinais de pulso, onde todos os canais são transmitidos juntos, e cada canal é determinado pela largura. Sinal PPM com seis canais do controle é apresentado na Figura 33.

Figura 33 – PPM.



Fonte: Próprio Autor (2024).

A alimentação do controle é 6V, sendo este feito por 4 pilhas do tipo AA, pesa 392 gramas e suas dimensões são 174\*89\*190 mm. Possui um pequeno display 128\*64 pixels, translucido com luz de fundo, com dimensões de 73\*39 mm. No display são exibidas informações ao usuário como nível das baterias e também taxa de erro do RF na troca de informações entre o controle e o receiver.

O controle possui três chaves seletoras de duas posições sendo estas SWA, SWB e SWD, e uma chave seletora de três posições SWC, que podem ser usadas para configurar comandos de resposta para o drone como, por exemplo, o modo retorno automático (RTL), ainda conta com dois potenciômetros VRA e VRB, que podem ser usados para variar parâmetros em voo, como por exemplo, o ganho de PID que controla a resposta o comando de direção, podendo ser bem úteis na hora de configurar ganhos de parâmetros.

Para receber os comandos do controle é preciso de um receiver, deste conjunto é o FS-IA10B, sendo este compatível com os dez canais de RF do controle, e tem um alcance de até 1500 metros, este possui dois fios que saem do receiver que são as suas antenas. Ele possui um barramento com onze saídas, uma para cada canal, mais uma que serve para fazer a sincronia entre o controle e o receiver, uma vez sincronizados os dois, não é mais usada essa saída.

Cada saída tem três pinos, sendo elas, Sinal, VCC e GND. No trabalho é utilizada apenas a saída numero um. Sendo esta a saída do PPM, então saindo do pino um o Sinal PPM dos comandos recebidos do controle para a controladora, e os outros dois pinos serão responsáveis pela alimentação do receiver, sendo eles respectivamente VCC e GND. O receiver é apresentado na Figura 34.

Figura 34 – Receiver FS-IA10B.



Fonte: Próprio Autor (2024).

### 3.2.5 Frame

O frame é o corpo principal do drone, sua estrutura precisa ser resistente, tendo isso em vista foi selecionado o kit ZD680 da marca TJI, feito de fibra de carbono. Sua base tem a dimensão de 680 mm, as hastes de sustentação dos motores têm 260 mm de comprimento.

Esse modelo tem braços móveis, quando necessário para transporte, por exemplo, podem-se dobrar as hastes dos motores, diminuindo o tamanho da estrutura tomando-a mais fácil de transportar. O frame apresentado na Figura 35.

Figura 35 – Frame.



Fonte: Próprio Autor (2024).

### 3.2.6 Motores

Os motores selecionados foram do tipo Brushless, modelo 5010 da marca FlyCat, esses motores tem ímãs de Neodímio, Ferro e Boro (NdFeB), contando com rolamentos de esferas blindadas, são capazes de produzir até 1.500 gramas de empuxo com uma bateria de 4S. Sua dimensão é de 50\*50\*10 mm, com um peso de 80 gramas, necessitam de um controle eletrônico de velocidade (ESC) pra seu correto funcionamento, seu melhor rendimento é com hélices de 14 a 16 polegadas, a medida de potência destes motores é KV que representam sua capacidade máxima de rotação, é obtida pela seguinte equação.

$$KV = \frac{rpm}{V} \quad (3)$$

Os motores do tipo Brushless de corrente contínua (DC), são muito utilizados em drones e aeromodelos em geral por apresentar uma considerável potência e resistência com um peso e tamanho consideravelmente pequeno, o que é muito importante em equipamentos de voo, pois quanto mais peso for adicionado à estrutura do drone maior será a força necessária para mantê-lo no ar, conseqüentemente maior será o consumo da bateria, uma vez que os motores transformam potencial elétrico em movimento mecânico, e desta forma o consumo acaba diminuindo a autonomia do conjunto.

É preciso ressaltar que drones tendem a exigir bastante das baterias e dos motores, uma vez que é necessária a constante compensação da força da gravidade, e adicionado a esse esforço constante a compensação do vento, uma vez que o drone precisa manter-se estável em relação ao seu eixo XYZ. Na Figura 36 é apresentado o motor.

Figura 36 – Motores.

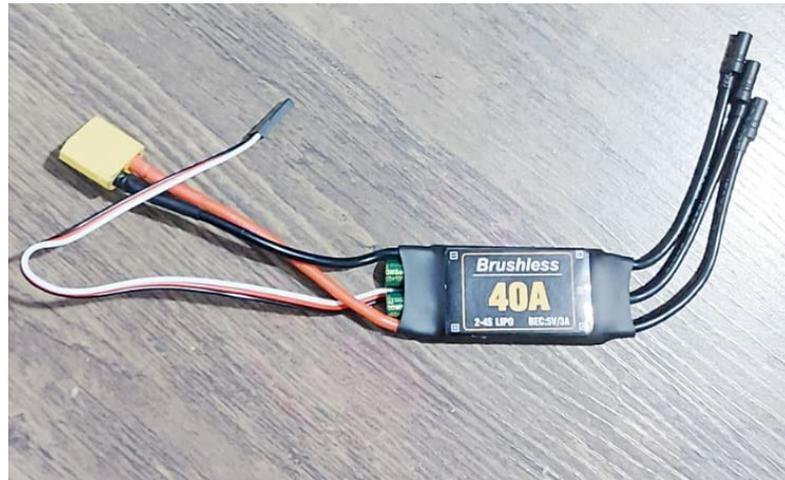


Fonte: Próprio Autor (2024).

### 3.2.7 ESC

Os motores do tipo Brushless tem uma alta demanda de corrente por isso necessitam de uma placa do tipo ESC, que suporta e sustenta essa demanda, além disto, controla a velocidade e também indica o lado correto de rotação, são controladoras do tipo PWM, para o projeto foi selecionado um modelo de 40A da marca Hobbywing conforme a Figura 37 mostra.

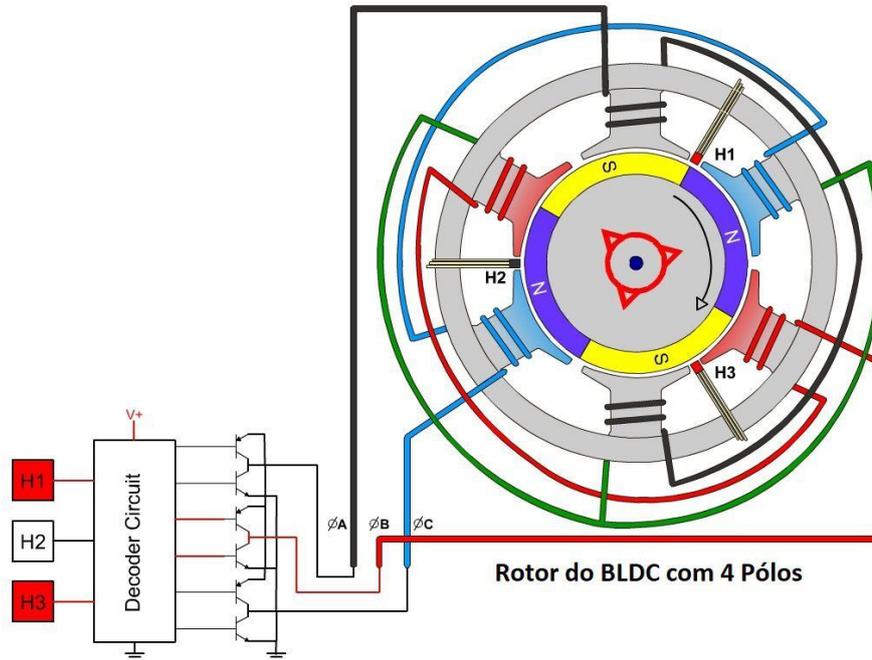
Figura 37 – ESC.



Fonte: Próprio Autor (2024).

Outra função do ESC é alimentar corretamente o motor, ou seja, é preciso fazer a comutação das bobinas em uma determinada ordem para que os motores girem corretamente. Na Figura 38 é apresentado o controle de acionamento.

Figura 38 – ESC controle de acionamento.



Fonte: Kalatec (2024).

### 3.2.8 Telemetria

A telemetria será feita por dois módulos RF da marca 3DR, esses módulos se comunicam a uma frequência de banda de 915 Mhz com uma potência de 500mW, tem uma taxa de dados no ar de 250 kbps, seu alcance aproximado é de 1,5 km, mas pode ser aumentado com uma antena do tipo omni. O peso do conjunto completo é de 63 gramas, o módulo do ar tem as dimensões de 51\*30\*10mm/12g e o módulo terra tem as dimensões de 70\*25\*12mm/15g. Na Figura 39 são apresentados os módulos de telemetria.

Figura 39 – Telemetria.



Fonte: Próprio Autor (2024).

### 3.2.9 BEC

A placa de distribuição de energia do projeto é o modelo CC3D V2 ZMR da marca Espektrum, ela é responsável por distribuir a energia aos componentes do drone e também por manter constante a tensão para alguns componentes que precisem de uma alimentação constante como, por exemplo, LED's de sinalização que podem ser colocados no drone para voos noturnos. Na Figura 40 é apresentada a placa BEC.

Figura 40 – BEC.



Fonte: Próprio Autor (2024)

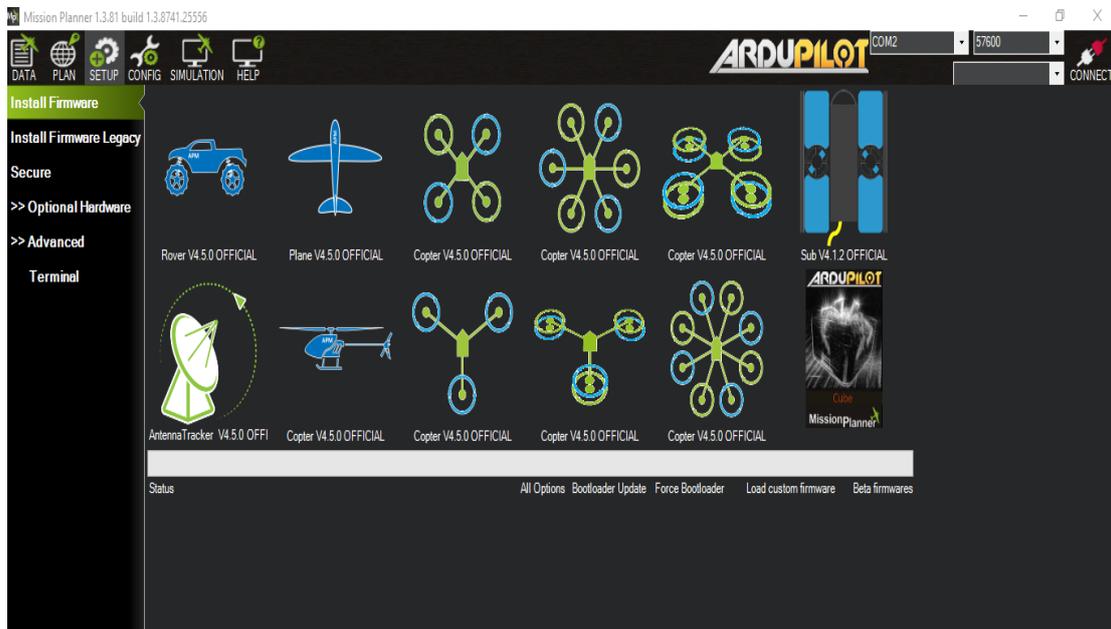
### 3.3 FIRMWARE

O firmware que vai gravado no cartão de memória da placa controladora é o PX4, em sua página web ele é definido como um piloto automático de código aberto, estando seu código fonte disponível em uma página na internet, ou seja, qualquer pessoa pode ver e também pode fazer alterações, sugerir melhorias, etc.

A instalação do firmware é feita pelo próprio programa de controle Mission Planner, onde é necessário conectar a placa controladora por meio de um cabo USB, o programa reconhece à controladora de forma automática, caso ele não reconheça a controladora o usuário pode selecionar o modelo de controladora ou uma controladora genérica, então o programa oferece opções de atualização caso existam, e em seguida o usuário escolhe o tipo de veículo que deseja controlar. Isso ainda sem clicar no botão de CONNECT existente no canto superior direito, pois uma vez que faça essa conexão entra no software da controladora e o firmware fica o que já está instalado dentro da placa, logo só será possível fazer alterações de parâmetros específicos do modelo controlado. Uma vez selecionado o modelo desejado, no caso do trabalho será o Copter V4.5.1 OFFICIAL de 4 motores, então o programa irá gravar o firmware no cartão SD da controladora.

Após a gravação bem sucedida do firmware no cartão é possível clicar no botão CONNECT no canto superior direito e alterar parâmetros específicos da modelo selecionado bem como realizar leituras, testes e gravação. Na Figura 41 é apresentada a tela de gravação do firmware.

Figura 41 – Tela de gravação do Firmware.



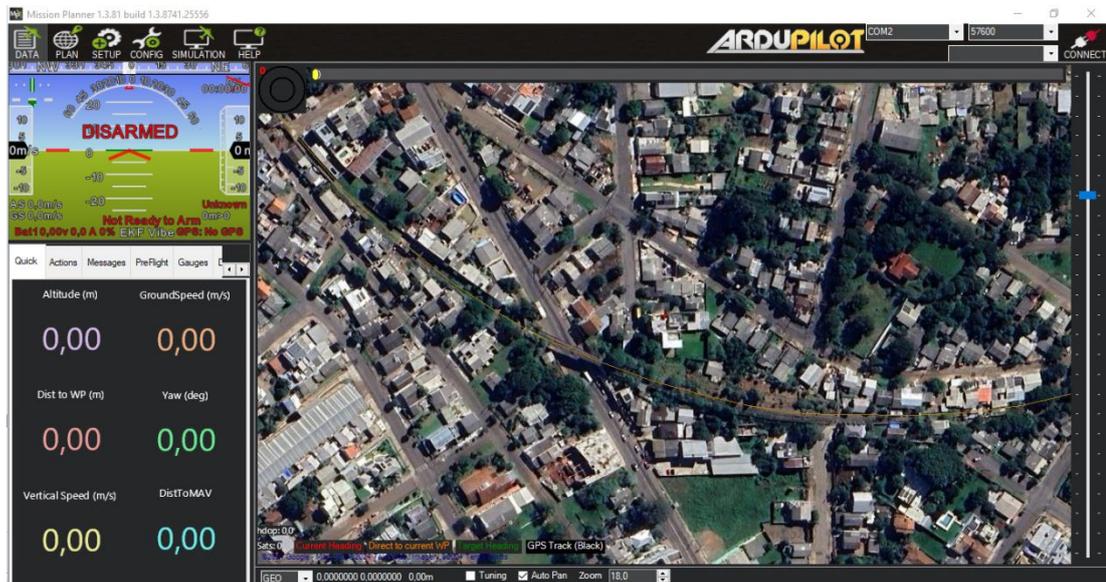
Fonte: Próprio Autor (2024).

### 3.4 SOFTWARE

O software que trabalha em conjunto com essa controladora é o Mission Planner, sendo do tipo código aberto. Ele conta com a participação de vários desenvolvedores e foi inicialmente fundado por Michael Osborne, contém uma vasta gama de opções de controle e configuração. Através do Mission Planner é feita as configurações da controladora desde a gravação do firmware, a seleção da estrutura de física do drone até as configurações avançadas. E por meio deste que será feita a calibração de sensores como GPS, barômetro entre outros e posteriormente a leitura de dados de voo, por meio dele será realizada a função de seguir outro dispositivo móvel.

Esse software é compatível com inúmeras placas controladoras, desde as fabricadas por empresas parceiras do projeto, sendo estas definidas como modelos oficiais, ou com placas de empresas que copiaram o modelo e fizeram sua própria sendo essas definidas como genéricas. Na Figura 42 é apresentada a tela inicial do Mission Planner.

Figura 42 – Mission Planner tela inicial.



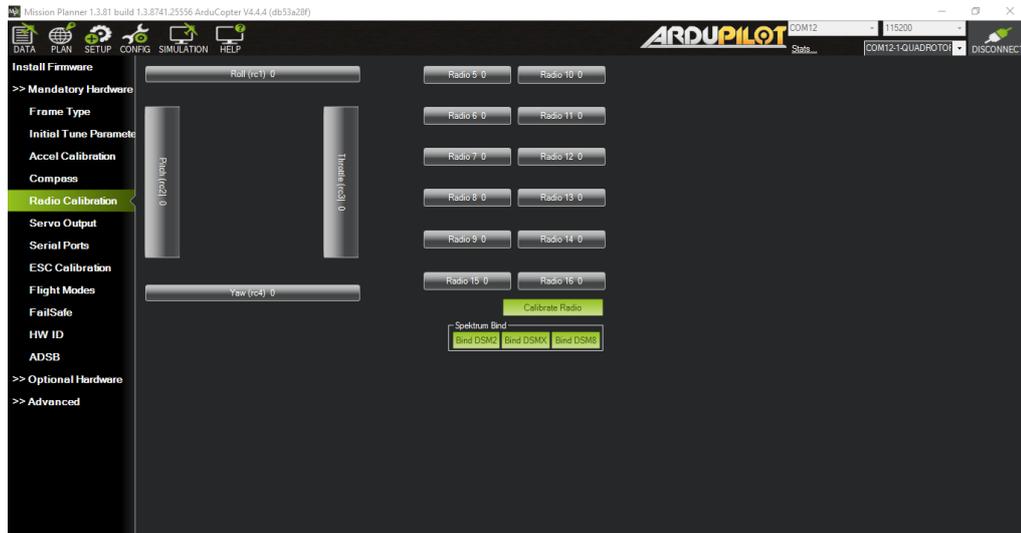
Fonte: Próprio Autor (2024).

Na tela inicial são apresentadas algumas informações sobre o drone, como a posição do GPS em um mapa, no canto esquerdo superior é apresentada uma leitura da controladora com a posição atual em relação ao solo, e no canto inferior as informações como altura, velocidade, elevação, esse último é personalizável, podendo o usuário alterar quais os parâmetros que serão exibidos, clicando na aba de seleção pode escolher entre Quick, Actions, Messages etc. No canto superior esquerdo encontram-se os menus do programa sendo eles Data, Plan, Setup, Config Simulation e Help. No canto superior direito encontra-se o botão CONNECT e as opções de conexão como a seleção da porta e da velocidade.

Para as configurações iniciais é preciso conectar o cabo USB e após olhar no canto superior direito e ver se já aparece a porta COM que vai fazer a comunicação com a controladora, se já estiver ok basta clicar no botão CONNECT para fazer a conexão, após isso abrir o menu de seleção Setup, dentro do mesmo encontram-se as opções, de instalar o firmware, as opções de hardware e as opções avançadas.

Na opção de Mandatory Hardware tem todas as configurações obrigatórias, como a seleção do tipo de Frame, a configuração parâmetros iniciais, a calibração de sensores, calibração de radio controle, a calibração dos motores, a calibração dos módulos ESC entre outras. Na Figura 43 é apresentada a tela de calibração do radio controle.

Figura 43 - Tela de calibração do rádio.



Fonte: Próprio Autor (2024).

Pode-se notar nesta tela que ela exibe todos os canais de comunicação com rádio que a controladora suporta, sendo o máximo de 16 canais, os quatro primeiros canais são dedicados ao controle manual do drone, ou seja, ao sticks do controle sendo eles canal 1 ROLL, canal 2 PITCH, canal 3 THROTTLE e canal 4 YAW. E os demais são utilizados para modos de voo (Flight Modes), sendo que as funções dos canais acima dos quatro são personalizáveis, ou seja, o usuário pode escolher o que deve ser feito caso ele esteja ativado ou desativado.

O controle selecionado para este projeto inicialmente transmitia em seis canais de comunicação, sendo este o mínimo que é necessário para o correto funcionamento da placa como citado anteriormente são necessários quatro canais para o controle manual do drone, um canal é reservado para a placa comunicar com a estação terrestre e logo sobrava apenas um canal para selecionar os modos de voo, como o controle conta com três chaves de duas posições e apenas uma chave de três posições, o máximo que seria possível selecionar de modos de voo seria três. Porém este controle pode trabalhar com até 10 canais de rádio comunicação, e para isso foi necessário fazer uma atualização do controle, com a atualização do controle passou a ter mais canais de comunicação.

### 3.4.1 Follow me

A função de seguir outro dispositivo é implementada por software, o código usado de base pode ser visto no Anexo A, primeiramente é preciso comunicar com o Mission Planner e para isso é preciso usar as bibliotecas específicas, também é preciso fazer algumas definições, abrir portas de comunicação, para que possa ser selecionado o meio de comunicação, e a taxa de atualização da porta.

#### Principais Ações do Código:

1. Carrega Configurações: Carrega portas seriais e define a taxa de atualização.
2. Evento de Clique no Botão "Conectar".
3. Conecta ao Drone: Estabelece conexão com o drone.
4. Obtém Altitude: Solicita altitude relativa ao usuário.
5. Inicia Thread: Inicia a thread principal para seguir o usuário.
6. Processa Dados NMEA: Lê e processa dados de localização NMEA.
7. Envia Waypoints: Envia comandos de waypoint ao drone para seguir a localização.
8. Atualiza Interface: Atualiza a interface com a localização atual do drone.
9. Calcula Checksum: Verifica a integridade dos dados NMEA.
10. Trata Alteração de Taxa: Ajusta a taxa de atualização dos comandos.

Na Figura 44 são apresentadas as definições da comunicação.

Figura 44 – Comunicação.

```

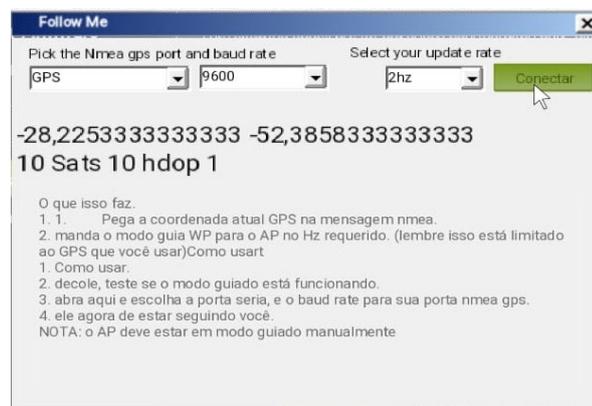
43 // Inicializa os componentes do formulário
44 InitializeComponent();
45
46 // Carrega a lista de portas seriais disponíveis
47 CMB_serialport.DataSource = SerialPort.GetPortNames();
48
49 // Define a taxa de atualização padrão
50 CMB_updaterate.SelectedItem = updaterate;
51
52 // Se a thread estiver em execução
53 if (threadrun)
54 {
55     // Altera o texto do botão para "Parar"
56     BUT_connect.Text = Strings.Stop;
57
58     // Define a taxa de baudrate
59     CMB_baudrate.Text = comPort.BaudRate.ToString();
60
61     // Define a porta serial
62     CMB_serialport.Text = comPort.PortName;
63
64     // Define a taxa de atualização
65     CMB_updaterate.Text = updaterate.ToString();
66 }

```

Fonte: Ardupilot adaptado (2024).

Em caso de uma conexão bem sucedida é exibido ao usuário uma caixa de diálogo solicitando a altura em metros que ele deseja que o drone fique, com uma sugestão de 10 metros de altura, após o usuário imputar o valor é iniciada a função de seguir (Follow me), e então são exibidas na caixa de diálogo as mensagens de latitude e longitude, juntamente com algumas informações ao usuário. Na Figura 45 é apresentada a caixa de diálogo *Follow me*.

Figura 45 – Caixa de dialogo.



Fonte: Proprio autor (2024).

Feito isso é entrado no loop principal do programa com a execução do *follow me* propriamente dita o código para a função é relativamente grande então é apresentado apenas às partes mais significativas. O bloco *try* é feito para lidar com possíveis exceções que podem ocorrer durante a execução do código dentro do *loop*, no tratamento das mensagens NMEA. Isso garante que o programa não trave caso haja algum erro inesperado. Na Figura 46 é apresentado void loop.

Figura 46 – Void loop.

```

187 // Laco principal do Follow Me
188 while (threadrun)
189 {
190     try
191     {
192         // Lê uma linha de dados NMEA da porta serial
193         string line = comPort.ReadLine();
194
195         // Grava a linha de dados NMEA no arquivo
196         sw.WriteLine(line);
197
198         // Verifica se a linha de dados NMEA é um NMEA GGA ou GNGGA
199         if (line.StartsWith("$GPGGA") || line.StartsWith("$GNGGA"))
200         {
201             // Divide a linha de dados NMEA em tokens
202             string[] items = line.Trim().Split(',');
203
204             // Verifica se o checksum da linha de dados NMEA é válido
205             if (items[items.Length - 1] != GetChecksum(line.Trim()))
206             {
207                 // Se o checksum não for válido, ignora a linha
208                 Console.WriteLine("Bad Nmea line " + items[15] + " vs " + GetChecksum(line.Trim()));
209                 continue;
210             }
211         }

```

Fonte: Ardupilot adaptado (2024).

É feito a leitura da NMEA na porta serial do dispositivo e então gravado em um arquivo a leitura, após é feito uma checagem para saber qual o tipo de mensagem é GGA ou GNGGA, também se verifica se a mensagem é válida, caso seja inválida o programa ignora a linha e faz uma nova captura, caso seja validada a mensagem ele atualiza a string e divide os dados em token, cada token representa um campo específico da mensagem NMEA.

Feito uma verificação da validade do checksum de dados NMEA, o checksum tem um valor de redundância que garante a integridade dos dados transmitidos. A função GetChecksum calcula o valor esperado e o compara com o último token, se os valores não coincidirem, a linha é considerada inválida e ignorada. Caso a leitura seja válida então é hora de ir para os dados da mensagem. Na Figura 47 é apresentado o parte do código para os dados.

Figura 47 – Dados da NMEA.

```

212 // Verifica se há fixação de GPS
213 if (items[6] == "0")
214 {
215     // Se não houver fixação de GPS, ignora a linha
216     Console.WriteLine("No Fix");
217     continue;
218 }
219
220 // Atualiza a latitude do local de destino
221 gotolocation.Lat = double.Parse(items[2], CultureInfo.InvariantCulture) / 100.0;
222 gotolocation.Lat = (int)gotolocation.Lat + ((gotolocation.Lat - (int)gotolocation.Lat) / 0.60);
223
224 // Verifica se a latitude é sul
225 if (items[3] == "S")
226 {
227     gotolocation.Lat *= -1;
228 }
229
230 // Atualiza a longitude do local de destino
231 gotolocation.Lng = double.Parse(items[4], CultureInfo.InvariantCulture) / 100.0;
232 gotolocation.Lng = (int)gotolocation.Lng + ((gotolocation.Lng - (int)gotolocation.Lng) / 0.60);
233
234 // Verifica se a longitude é oeste
235 if (items[5] == "W")
236 {
237     gotolocation.Lng *= -1;
238 }
239
240 // Define a altitude do local de destino
241 gotolocation.Alt = intalt;
242
243 // Atualiza a tag do local de destino com informações de satélite
244 gotolocation.Tag = "Sats " + items[7] + " hdop " + items[8];
245

```

Fonte: Ardupilot adaptado (2024).

É verificado se o sexto token é válido, caso ele seja igual à zero indica que não há fixação de GPS e a linha é ignorada. Quando estiver correta então é extraída a latitude do local de destino do segundo token, convertem-na para um número de ponto flutuante, em

seguida, divide o valor por 100 para converter de graus para centésimos de grau, que é a unidade de medida comum para a longitude em dados NMEA.

É verificado se o quinto token da linha NMEA é igual a "W", que indica que a longitude é oeste. Se for verdade, o código multiplica a longitude por -1 para inverter o sinal e representar a longitude oeste corretamente, o mesmo acontece se este for "S", indicando latitude sul.

Ao usar coordenadas geográficas para controlar o movimento de um drone, é crucial representar as direções Norte e Sul corretamente. No entanto, algumas linguagens de programação ou bibliotecas podem armazenar a latitude como um valor numérico simples, sem considerar a direção (Norte ou Sul).

Para garantir que o drone interprete as latitudes corretamente e siga o local de destino com precisão, é necessário inverter o sinal da latitude quando ela for sul e/ou oeste. Isso é feito multiplicando a latitude e/ou longitude por -1. Por exemplo, ao multiplicar -37.783333 por -1, obtemos 37.783333. Isso garante que o drone interprete a latitude como 37.783333° Sul e voe na direção correta. Na Figura 48 é apresentado o envio da posição de destino.

Figura 48 – Envio do destino.

```

256 // Armazena o local de destino anterior
257 lastgotolocation = new PointLatLngAlt(gotolocation);
258
259 // Cria um novo objeto Locationwp para representar o waypoint
260 Locationwp gotohere = new Locationwp();
261
262 // Define o tipo de comando como waypoint
263 gotohere.id = (ushort)MAVLink.MAV_CMD.WAYPOINT;
264
265 // Define a altitude do waypoint
266 gotohere.alt = (float)(gotolocation.Alt);
267
268 // Define a latitude do waypoint
269 gotohere.lat = (gotolocation.Lat);
270
271 // Define a longitude do waypoint
272 gotohere.lng = (gotolocation.Lng);
273
274 try
275 {
276     // Atualiza o rótulo do formulário com as informações do waypoint
277     updateLocationLabel(gotohere);
278 }

```

Fonte: Ardupilot adaptado (2024).

Após a atualização esses dados são passados para uma strig chamada *lastgotolocation*, que será usada para comparar se houve mudança na posição atual para a anterior. É criado

um objeto para a nova posição a ser enviada na string *gotohere*, que indica ao drone que a localização enviada deve ser seguida.

Após tem um bloco *try* que garante que qualquer erro que possa acontecer seja ignorado, e atualiza a interface de dados com o usuário, e então é feito o envio dos dados. Na Figura 49 é apresentado o envio da posição de destino.

Figura 49 – Envio do destino erros.

```

274 | try
275 | {
276 |     // Atualiza o rótulo do formulário com as informações do waypoint
277 |     updateLocationLabel(gotohere);
278 | }
279 | catch
280 | {
281 |     // Ignora qualquer erro que possa ocorrer ao atualizar o rótulo
282 | }
283 |
284 | // Se a porta serial estiver aberta e o drone estiver em modo de controle manual
285 | if (MainV2.comPort.BaseStream.IsOpen && MainV2.comPort.giveComport == false)
286 | {
287 |     try
288 |     {
289 |         // Envia o waypoint para o drone como um comando guiado
290 |         MainV2.comPort.setGuidedModeWP(gotohere, false);
291 |     }
292 |     catch
293 |     {
294 |         // Ignora qualquer erro que possa ocorrer ao enviar o waypoint
295 |     }
296 | }

```

Fonte: Ardupilot adaptado (2024).

Vale destacar a função de *checksum*, que calcula o valor hexadecimal de uma linha de dados NMEA que é usada para validar a integridade dos dados recebidos, dentro do *switch* é percorrido cada caractere da linha então através de uma operação XOR que combina os bits do caractere atual com o valor do *checksum*, alterando os bits que diferem e mantendo os bits iguais, após terminar o valor é convertido para hexadecimal e retornado na função.

Com isso são interpretados os dados de uma fonte GPS, e feito o tratamento desses dados, para posterior enviar ao drone a posição para deslocamento.

## 4 RESULTADOS E DISCUSSÃO

Neste capítulo está a apresentação dos resultados do trabalho. A montagem do drone necessitou de várias etapas, começando pela montagem do frame. O problema inicial de montar o frame foi pela falta de manual de montagem, por se tratar de uma peça que apresenta um volume considerável para transporte ela vem totalmente desmontada, contando com uma quantidade significativa de componentes. Na Figura 50 é apresentada a forma que chegou o frame.

Figura 50 – Frame desmontado.



Fonte: Próprio Autor (2024).

Após a montagem do frame é começada a etapa de fixação e conexão dos componentes, onde foi iniciado pelos motores nas pontas dos braços. Posteriormente foram soldados na placa de distribuição de energia BEC, todos os componentes que precisam de alimentação direta da bateria, logo foi soldado um cabo que vem diretamente da bateria para a placa de distribuição BEC, e foram soldados cabos para alimentar os ESC, além de cabos para alimentar LED's de sinalização e uma saída auxiliar de 12 V que não foi utilizada no

projeto. A alimentação da controladora não vem da placa de distribuição, ela é alimentada diretamente pela bateria. Na Figura 51 é apresentado o frame montado.

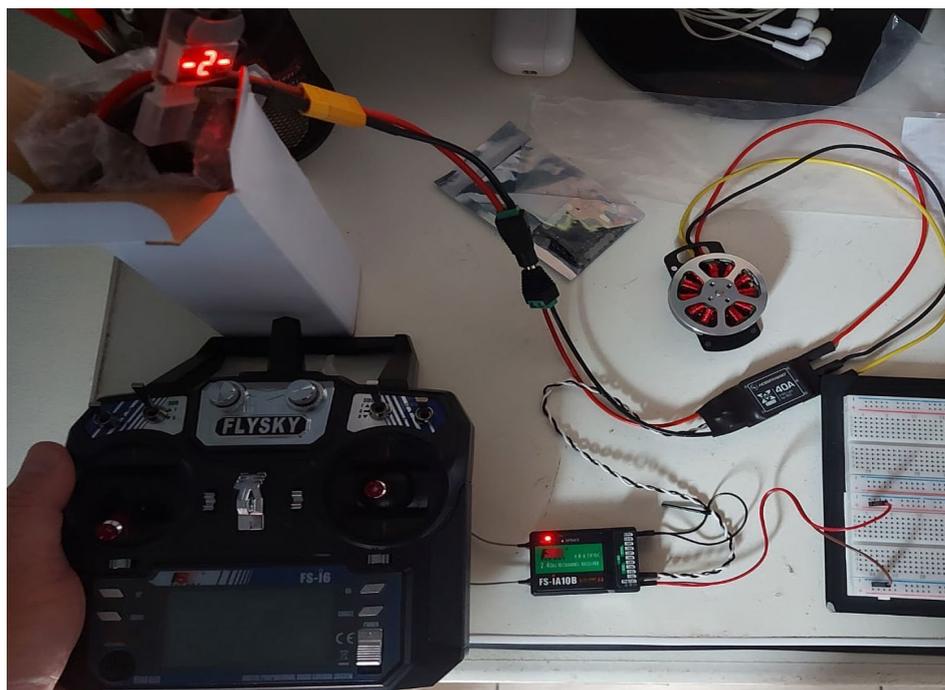
Figura 51 – Frame montado.



Fonte: Próprio Autor (2024).

Claro que os componentes foram testados antes de serem fixados, os motores e os BEC's foram testados fazendo uma conexão da bateria para o receiver do controle, e o receiver ligado ao BEC que era ligado ao motor, onde após receber dados do controle remoto o receiver enviado ao BEC dados de aceleração, e já que os motores são os maiores consumidores de corrente do conjunto, a bateria foi testada ao mesmo tempo. Na Figura 52 é apresentada a montagem para testes.

Figura 52 – Testes.



Fonte: Próprio Autor (2024).

A placa controladora também foi testada, onde foi conectada ao GPS, buzzer, chave de segurança, e o receiver, após o reconhecimento dos sensores foi feito a calibração dos sensores, onde é preciso, por exemplo, girar o conjunto em diferentes direções para o reconhecimento dos valores corretos de direita, esquerda, acima e a baixo, além da altura em relação ao solo.

A comunicação via rádio precisa ser configurada, é o chamado *bind* é um pareamento do controle com o receiver, e após é feita a gravação desses parâmetros tanto no rádio como no receiver.

O rádio controle também precisa de uma calibração específica com a controladora para que todos os canais sejam configurados com seus máximos e mínimos dos botões, além da escolha do que os canais auxiliares devem fazer.

Foi necessário atualizar o controle para que o mesmo trabalhasse com dez canais ao invés de apenas seis como configurados de fábrica. Com a atualização do controle também passou a ser possível configurar condições de resposta para um canal, por exemplo, SWA + SWB, onde o drone só executara o comando caso as duas chaves sejam ativadas, isto não era possível antes de fazer a atualização.

A conclusão desta etapa é a estrutura do drone pronta conforme pode ser visto na Figura 53.

Figura 53 – Drone.



Fonte: Próprio Autor (2024).

Com o drone montado é hora de configurar o software e após testar em voo, primeiro é preciso fazer alguns ajustes como informar o tipo de bateria e quantidade de células. Se o usuário deseja monitorar ou não a tensão da bateria.

Outro passo importante é o primeiro voo, é preciso que o usuário não decole e leve o drone para voar alto, pois muitos parâmetros ainda não foram ajustados, é preciso que o usuário decole o drone a uma altura máxima de dois metros e logo pouse para que o controlador ajuste alguns parâmetros iniciais como a resposta dos motores e configurações de PID, por exemplo, e após o pouso é possível gravar esses parâmetros para melhor funcionamento do drone.

Com o drone voando corretamente de forma estável e segura, então a parte final do trabalho é fazer a conexão entre o drone e o dispositivo eletrônico, neste caso foi feito com um celular. O celular é da marca Samsung modelo A71, a opção por utilizar um celular é a praticidade de ele já ter um GPS incorporado e sendo apenas necessário conectar a telemetria feita via rádio, abrir a porta de conexão e enviar ao drone a posição instantânea do celular para que seja feito o acompanhamento, além do celular e do rádio transmissor foi necessário apenas um adaptador de USB para o USB-C, padrão deste modelo de celular. Na Figura 54 é apresentado o celular conectado a telemetria.

Figura 54 – Celular e telemetria.



Fonte: Próprio Autor (2024).

Com a estrutura toda pronta e software devidamente configurado, foi preciso um local seguro para testes, para isso foi usado o campus da UPF, visto que o drone ficou com um peso considerável, e era preciso uma área grande como um campo de futebol, e sem pessoas próximas, visando minimizar o risco de acidentes envolvendo terceiros durante os testes. Nas Figuras 55, 56 e 57 são apresentados os testes do drone.

Figura 55 – Drone em teste.



Fonte: Próprio Autor (2024).

Figura 56 – Drone em teste 2.



Fonte: Próprio Autor (2024).

Figura 57 – Drone em teste 3.



Fonte: Próprio Autor (2024).

Com o final do projeto foi feito uma tabela com o levantamento dos gastos para desenvolver o projeto. Na Tabela 2 os gastos do projeto.

Tabela 2: Gastos

Item	Valor
Telemetria	256,87
ESC	141,43
Cabo Flysky	48,99
Câmera	191,50
USB Adapte	18,37
Carregador M7	128,38
Baterias	505,43
Modulo T-Beam	230,18
Drone DIY	3.286,96
Total	4808,11

Fonte: Próprio autor (2024).

## 5 CONSIDERAÇÕES FINAIS

Com o término do projeto, é possível comprovar que foi cumprido o objetivo inicial proposto pelo autor, embora tenha sido comprovado pelo mesmo que o desenvolvimento e montagem de dispositivos que voam são extremamente difíceis e exigem conhecimento por parte do desenvolvedor.

É preciso bastante pesquisa sobre os componentes, sendo que cada um deles é importante, e não existe muito material sobre o assunto.

Durante o desenvolvimento do projeto houve incidentes, como a queima de um dos ESC fazendo com que o drone sofresse uma queda de uma altura considerável quebrando um dos suportes do trem de pouso, foi necessário à compra de um novo conjunto de controladoras ESC e fazer um remendo no trem de pouso para que o drone pudesse aterrissar.

Um grande problema de se construir drones, aviões e carros do tipo DIY é que não há opções de compra no Brasil, fazendo-se necessária a importação de praticamente todas as peças, acarretando problemas como o tempo de chegada, a burocracia da alfandega, as taxas de importação, e o risco de que os componentes não sejam de boa qualidade.

Quanto ao projeto finalizado notou-se que a pilotagem de drones é algo bastante divertido e que dependendo do peso do conjunto é possível uma boa autonomia de voo, sendo que neste projeto a autonomia ficou em algo próximo de 1 hora de voo.

A escolha de um conjunto de fibra de carbono também se mostrou bastante assertiva, uma vez que resistiu a queda de uma boa altura e a única parte que quebrou foi uma conexão que era de plástico. Um ponto de melhoria seria construir uma capa protetora pra os componentes como a controladora, por exemplo, visto que os componentes ficam bastante expostos e podem sofrer danos em caso de quedas.

A telemetria via rádio mostrou-se bastante problemática, alternando entre horas funcionando bem e horas não, necessitando de constantes desligamentos para que reinicialize e realizasse nova conexão entre a estação terrestre e o drone, porém ela é fundamental no controle automático do drone e também para o melhor funcionamento do mesmo, uma vez que ficou muito claro pelos testes que quando o drone tem comunicação com a estação terrestre ele tem uma maior estabilidade.

Os motores se mostraram bastante confiáveis, não apresentando nenhum problema. O mesmo discurso não pode ser aplicado aos controladores ESC, que apresentaram problemas. Um específico controlador aqueceu e queimou levando a uma queda do drone.

A controladora tem muitos parâmetros configuráveis e exige muito estudo e entendimento por parte do usuário, a própria interface do programa não é muito intuitiva, talvez pelo fato de ter compatibilidade com muitos modelos de drones e aeromodelos, além de outros tipos de veículos, e em adição a isso ainda existe uma grande gama de opções de controle no programa como voo autônomo e simulações de voo.

Existem inúmeros parâmetros que podem ser configurados pelo usuário na lista de comandos avançados, mas eles exigem pesquisa por parte do usuário para entender qual alteração esta sendo feita.

Em suma o trabalho comprovou que o a construção de aeromodelos exige muita pesquisa e conhecimento além de ter bastante complexidade de controle, no ponto de vista do autor construir drones é válido para pesquisa e desenvolvimento, caso seja para uso recreativo o autor recomenda a compra de um modelo comercial.

Em relação ao desenvolvimento do sistema "*follow me*", o autor comprovou a sua implementação, destacando, entretanto, a existência de um amplo campo de possibilidades de melhorias que podem ser exploradas e desenvolvidas.

## REFERÊNCIAS

- AGUIAR, Luiz Fernando Ramos. **Drones entram de vez na atuação dos policiais**. Piloto Policial, 2017. Disponível em: < <https://www.pilotopolicial.com.br/drones-entram-de-vez-na-atuacao-dos-policiais/>>. Acesso em: 09 out. 2023, 22:16:19.
- ALECRIM, Emerson; HIGA, Paulo. **Qual é a diferença entre GPS, GLONASS e Galileo? Entenda os sistemas de navegação via satélite**. Tecnoblog, 2023. Disponível em: <https://tecnoblog.net/responde/diferenca-gps-glonass-galileo/>. Acesso em: 28 out. 2023, 20:07:32.
- ALVARADO, Ed. **Drone Market analysis 2022-2030**, Drone industry insights, 2022. Disponível em: <https://droneii.com/drone-market-analysis-2022-2030>. Acesso em: 29 out. 2023, 00:17:51.
- ARDUPILOT Dev Team. **Gráfico Avançado de fiação do Quadcopter Pixhawk**. Disponível em: <https://ardupilot.org/copter/docs/advanced-pixhawk-quadcopter-wiring-chart.html>. Acesso em: 30 de mar. 2024, 20:19:00.
- BADDELEY, Glenn. **GPS - NMEA sentence information**. 2001. Disponível em: <https://aprs.gids.nl/nmea/>. Acesso em: 29 out. 2023, 13:20:00.
- BLOECHL, Mark. **SigFox vs LoRa: A comparison between technologies & business models**. Link Labs. Disponível em: <https://www.link-labs.com/blog/sigfox-vs-lora>. Acesso em 08 nov. 2023, 20:56:12.
- CARNEIRO, Bruno Barcelo. **Projeto de uma bancada experimental para avaliação do controle de um drone**. Projeto de conclusão de curso. Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia. Uberlândia, 2017.
- CARVALHO, Rogerio Muniz. **Comunicações analógicas e Digitais**. Vitoria - Es: Editado Pelo Autor, 2008. 398 p.
- DA REDAÇÃO. **Entenda a diferença entre Drone e VANT e suas aplicações práticas**. Techtudo, 2015. Disponível em: < <https://www.techtudo.com.br/noticias/2015/05/entenda-diferenca-entre-drone-e-vant-e-suas-aplicacoes-praticas.ghtml>>. Acesso em: 22 out. 2023, 10:36:32.
- DIYODE Magazine. **H4WK DIY Drone Kit. Build & Fly Your Own Quadcopter!** Kickstarter, 2018. Disponível em: <https://www.kickstarter.com/projects/1471974309/h4wk-diy-drone-kit-build-and-fly-your-own-quadcopt>. Acesso em: 26 out. 2023, 21:15:00.
- EMBARCADOS. **Conheça as principais tecnologias de comunicação sem fio**. Disponível em: <https://embarcados.com.br/tecnologias-de-comunicacao-sem-fio/>. Acesso em: 04 nov. 2023, 23:58:17.
- ESTES, Adam Clark. **Como começou essa história de transmitir informações sem fio**. Giz modo UOL, 2018. Disponível em: <https://gizmodo.uol.com.br/como-comecou-essa-historia-de-transmitir-informacoes-sem-fio/>. Acesso em: 02 nov. 2023, 00:18:40.

EXAME. **O detalhe que permitiu ao homem voar.** Revista Exame, 2017. Disponível em: <https://exame.com/colunistas/oportunidades-disfarcadas/o-detalhe-que-permitiu-ao-homem-voar/>. Acesso em : 06 nov. 2023, 21:15:00.

EXPLORERNET. **Wi-Fi: Entenda como surgiu a internet sem fio.** Blog Explorenet, 2022.

Disponível em: <https://explorenet.com.br/>. Acesso em: 02 nov. 2023, 00:51:10.

FLYPRO. **Anatomia de um Drone.** FlyPro, 2023. Disponível em: <https://www.flypro.com.br/pagina/anatomia-de-um-drone.html>. Acesso em: 22 out. 2023, 19:25:00.

FRAĆKIEWICZIN, Marcin. **Os melhores drones para esportes e fitness na Ucrânia.** 11 Março 2023. Disponível em: <https://ts2.space/pt/os-melhores-drones-para-esportes-e-fitness>. Acesso em: 13 nov. 2023, 18:34:18.

GARRETT, Filipe. **O que é Drone e para que serve?** Tectudo, 2015. Disponível em: <https://www.techtudo.com.br/noticias/2013/10/o-que-sao-e-para-que-servem-os-drones-tecnologia-invade-o-espaco-aereo.ghtml>. Acesso em: 14 out. 2023, 23:50:00.

GRANEMANN, Emerson. **Por que o mercado de drones não para de crescer?** Jornal O Presente Rural, 2023. Disponível em: <https://opresenterural.com.br/por-que-o-mercado-de-drones-nao-para-de-crescer>. Acesso em: 28 out. 2023, 23:42:15.

GROSS, Richard J. **Evolução completa & Historia dos Drones: De 1800s a 2022.** Propel, 2023. Disponível em: <https://www.propelrc.com/pt/historia-dos-fosseis/>. Acesso em: 09 out. 2023, 21:29:20.

GUITARRARA, Paloma. **GPS – Sistema de Posicionamento Global.** Brasil Escola, 2023. Disponível em: <https://brasilescola.uol.com.br/geografia/gpsistema-posicionamento-global.htm>. Acesso em: 28 out. 2023, 18:50:00.

HENTZ, Ângela Maria Klein. **Obtenção de Informações Dendrométricas para Inventário Florestal Automatizado por Meio de Veículo Aéreo Não Tripulado (VANT).** Tese (Doutorado em Engenharia Florestal) - Universidade Federal do Paraná, Curitiba, 2018.

INTEL. **Como funciona a tecnologia Bluetooth®?** Disponível em: <https://www.intel.com.br/content/www/br/pt/products/docs/wireless/how-does-bluetooth-work.html>. Acesso em: 08 nov. 2023, 11:18:23.

KAPLAN, E. D. HEGARTY, C. J. **Understanding GPS: principles and applications**, 3 ed. Artech House, Boston, 2017.

LATHI, B P.; DING, Zhi. **Sistemas de Comunicações Analógicas e Digitais Modernos, 4ª edição.** Grupo GEN, 2012. *E-book*. ISBN 9788521636076. Disponível em: <https://integrada.minhabiblioteca.com.br/>. Acesso em: 04 nov. 2023.

LEOSHMAN, J. Gordon. **Principles of Helicopter Aerodynamics.** Cambridge. 2000. Disponível em:

[https://assets.cambridge.org/97811070/13353/excerpt/9781107013353\\_excerpt.pdf](https://assets.cambridge.org/97811070/13353/excerpt/9781107013353_excerpt.pdf). Acesso em: 23 out. 2023, 20:43:00.

LEMMENS, Mathias. **O que é VANT? O que ele faz?**. DronEng, 2015. Disponível em: [O que é VANT? O que ele faz? - DronEng](#). Acesso em 13 nov. 09:58:02

MARTINS, Samuel. **Drones: o que são, como funcionam e como Voam**. Blog Eletrogate, 2022. Disponível em: <<https://blog.eletrogate.com/drones-o-que-sao-como-funcionam-e-como-voam/>>. Acesso em: 26 out. 2023, 20:31:00.

MEDIAWIKI. **WI-FI**, Instituto Federal Santa Catarina, Câmpus São José, 2007. Disponível em: <https://wiki.sj.ifsc.edu.br/index.php/Wi-Fi>. Acesso em: 07 nov 2023, 15:29:00.

MORGADO, Virginia Andújar. **Entorno de simulación en la banda ISM para test de algoritmos de detección espectral**. Projeto de conclusão de curso. Universidade Carlos III de Madrid. 2010. 111 f. Leganés, 2010.

MURTA, José Gustavo Abreu. **Conhecendo o ESP32 – Introdução**. BlogEletrogate, 2024. Disponível em: <https://blog.eletrogate.com/conhecendo-o-esp32-introducao-1>. Acesso em: 29 abril de 2024, 20:33:00.

NISHIKIORI, Igor. **De onde vem o nome Bluetooth? Veja significado, o que é e curiosidades**. Techtudo, 2022. Disponível em: <https://www.techtudo.com.br/noticias/2022/07/o-que-e-bluetooth-e-de-onde-vem-o-nome-veja-significado-e-curioidades.ghtml>. Acesso em: 06 nov. 2023, 21:43:00.

PANORAMA Positivo. **PROCESSAMENTO DE DADOS: Como fazer de maneira eficiente?** Disponível em: <https://www.meupositivo.com.br/processamento-de-dados/>. Acesso em: 05 nov. 2023, 01:20:00.

QUECTEL. **L70-R Series GPS: Protocol Specification**. [S.l.], 2016. 39 p. V3.0.

PASTÓRIO, A. F. ROSSATO, j. SÁ, F.A. Spanhol, LA Rodrigues, ET Camargo. **Fundamentos de LoRaWAN – Teoria e prática**. Trabalho de conclusão de curso. Sociedade Brasileira de Computação, 2021. 2021.

PREDROSA, Leyberson. **Drones, Vants ou RPAs? Entenda mais sobre essas aeronaves não tripuladas**. EBC, 2015. Disponível em: <https://memoria.ebc.com.br/tecnologia/2015/02/drones>. Acesso em: 22 out. 2023, 12:04:00.

RANGEL Fábio E, SILVA, Luciana da C, DIAS, Fábio F, ASSIS, Vânia C. de, PINTO, CAmille de B. Validação da acurácia posicional dos produtos resultantes do processamento digital de imagens de VANTs. REVISTA BRASILEIRA DE ENGENHARIA AGRÍCOLA E AMBIENTAL, 2022. Campina Grande-PB. 2022 p.1 – 7.

REVISTA ASAS. **Número de operações de drones no Brasil cresce 27,8% em um ano**. Revista Asas. 2023. Disponível em: <https://www.edrotacultural.com.br>. Acesso em: 28 out. 2023, 23:15:23.

ROBOCORE. **BlackBoard Edge H**. ESP32. Disponível em: <https://www.robocore.net/placa-robocore/blackboard-edge-h>. Acesso em: 05 nov. 2023, 19:23:00.

RODRIGUES, Lino. **Indústria de drones movimenta R\$ 300 milhões no Brasil**. ESTADO DE MINAS, 2018. Disponível em: <https://www.em.com.br/industria-de-drones-movimenta-r-300-milhoes-no-brasil.shtml>. Acesso em: 09 out. 2023, 20:46:30.

RUSSO, Rosana. **Conheça os principais componentes de um drone**. AEROJR, 2021. Disponível em: <https://aerojr.com/blog/principais-componentes-de-um-drone/>. Acesso em: 21 nov. 2023, 02:16:00.

SILVA, Jonas Fernandes da, ASSIS, Hugo Elias Gomes de, BRITO, Alisson V, ALMEIDA, Nadjacleia Vilar. X CONGRESSO BRASILEIRO DE AGROINFORMÁTICA. 2015. Ponta Grossa. **Anais...** Ponta Grossa: Universidade Estadual de Ponta Grossa. 2015 p. 1 – 10.

SIMONI, Gabriel Mendes. SCARAMELLA, Geovana. **Monitoramento de veículos alternativos**: Desenvolvimento de um protótipo com tecnologia LoRaWan. 2021. 150 f. Trabalho de conclusão de curso. Faculdade de Eletrônica, Universidade tecnológica Federal do Paraná. Curitiba, 2021.

SÓ FÍSICA. **GPS - O que é, como funciona**. *Só Física*. Virtuoso Tecnologia da Informação, 2008-2023. Disponível em: <http://www.sofisica.com.br/conteudos/curiosidades/gps.php>. Acesso em: 28 out. 2023, 21:58:00.

TEXAS INSTRUMENTS. **ARM® Cortex®-M4F-Based MCU TM4C1294 Connected LaunchPad™ Evaluation Kit**. Disponível em: <[EK-TM4C1294XL Evaluation board | TI.com](#)>. Acesso em: 05 nov. 2023, 01:20:00.

TEXAS INSTRUMENTS. **What is Zigbee?** Disponível em: <https://www.ti.com/zigbee>. Acesso em: 08 nov. 2023, 17:44:00

## ANEXO A – CÓDIGO FOLLOW ME

Código Follow me criado por Julian Oes

```
#include <iostream>
#include <cmath>
#include <thread>
#include <chrono>
#include <array>

#include "integration_test_helper.h"

#include "mavsdk.h"
#include "plugins/telemetry/telemetry.h"
#include "plugins/action/action.h"
#include "plugins/follow_me/follow_me.h"
#include "plugins/info/info.h"
#include "plugins/param/param.h"

using namespace mavsdk;
using namespace std::chrono;
using namespace std::this_thread;

/* Check if the autopilot supports improved follow me */
bool autopilot_has_improved_followme(const std::shared_ptr<Param> param);

/* Auxiliary Functions */
void print(const FollowMe::Config& config);

void send_location_updates(
    std::shared_ptr<FollowMe> follow_me, size_t count = 25ul, float rate = 2.f);

const size_t N_LOCATIONS = 100ul;
```

```

/* Test FollowMe with a stationary target at one location */
TEST(SitlTest, PX4FollowMeOneLocation)
{
    Mavsdk mavsdk {Mavsdk::Configuration{Mavsdk::ComponentType::GroundStation} };

    ConnectionResult ret = mavsdk.add_udp_connection();
    ASSERT_EQ(ConnectionResult::Success, ret);

    // Wait for system to connect via heartbeat.
    sleep_for(seconds(2));
    auto system = mavsdk.systems().at(0);
    ASSERT_TRUE(system->has_autopilot());

    auto telemetry = std::make_shared<Telemetry>(system);
    auto follow_me = std::make_shared<FollowMe>(system);
    auto action = std::make_shared<Action>(system);
    auto param = std::make_shared<Param>(system);

    // Skip the test if the Autopilot doesn't support the improved Follow Me (in MAVSDK
v2)
    if (!autopilot_has_improved_followme(param)) {
        GTEST_SKIP();
    }

    LogInfo() << "Waiting for system to be ready";
    ASSERT_TRUE(poll_condition_with_timeout(
        [telemetry]() {
            LogInfo() << "Waiting for system to be ready";
            return telemetry->health_all_ok();
        },
        std::chrono::seconds(10)));

    Action::Result action_ret = action->arm();
    ASSERT_EQ(Action::Result::Success, action_ret);
}

```

```

telemetry->subscribe_flight_mode([&follow_me](Telemetry::FlightMode flight_mode) {
    const FollowMe::TargetLocation last_location = follow_me->get_last_location();

    std::cout << "[FlightMode: " << flight_mode
                << "] Vehicle is at Lat: " << last_location.latitude_deg << " deg, "
                << "Lon: " << last_location.longitude_deg << " deg." << '\n';
});

action_ret = action->takeoff();
ASSERT_EQ(Action::Result::Success, action_ret);

sleep_for(seconds(5)); // let it reach takeoff altitude

auto curr_config = follow_me->get_config();
print(curr_config);

// Set just a single location before starting FollowMe (optional)
FollowMe::TargetLocation some_location;
some_location.latitude_deg = 47.39768399;
some_location.longitude_deg = 8.54564155;
some_location.absolute_altitude_m = 0.0;
some_location.velocity_x_m_s = 0.f;
some_location.velocity_y_m_s = 0.f;
some_location.velocity_z_m_s = 0.f;
follow_me->set_target_location(some_location);

// Start following with default configuration
FollowMe::Result follow_me_result = follow_me->start();
ASSERT_EQ(FollowMe::Result::Success, follow_me_result);
sleep_for(seconds(1));

std::cout << "We're waiting (for 5s) to see the drone moving target location set." << '\n';
sleep_for(seconds(5));

```

```

// stop following
follow_me_result = follow_me->stop();
ASSERT_EQ(FollowMe::Result::Success, follow_me_result);
sleep_for(seconds(2)); // to watch flight mode change from "FollowMe" to default
"HOLD"

action_ret = action->land();
ASSERT_EQ(Action::Result::Success, action_ret);
sleep_for(seconds(2)); // let the system land

while (telemetry->armed()) {
    std::cout << "waiting for system to disarm" << '\n';
    sleep_for(seconds(1));
}
}

/* Test FollowMe with a dynamically moving target */
TEST(SitlTest, PX4FollowMeMultiLocationWithConfig)
{
    Mavsdk mavsdk{Mavsdk::Configuration{Mavsdk::ComponentType::GroundStation}};

    ConnectionResult ret = mavsdk.add_udp_connection();
    ASSERT_EQ(ConnectionResult::Success, ret);

    // Wait for system to connect via heartbeat.
    ASSERT_TRUE(poll_condition_with_timeout(
        [&mavsdk]() {
            const auto systems = mavsdk.systems();

            if (systems.size() == 0) {
                return false;
            }
        }

```

```

        const auto system = mavsdk.systems().at(0);
        return system->is_connected();
    },
    std::chrono::seconds(10)));
auto system = mavsdk.systems().at(0);
ASSERT_TRUE(system->has_autopilot());

auto telemetry = std::make_shared<Telemetry>(system);
auto follow_me = std::make_shared<FollowMe>(system);
auto action = std::make_shared<Action>(system);
auto param = std::make_shared<Param>(system);

// Skip the test if the Autopilot doesn't support the improved Follow Me (in MAVSDK
v2)
if (!autopilot_has_improved_followme(param)) {
    GTEST_SKIP();
}

LogInfo() << "Waiting for system to be ready";
ASSERT_TRUE(poll_condition_with_timeout(
    [telemetry]() {
        LogInfo() << "Waiting for system to be ready";
        return telemetry->health_all_ok();
    },
    std::chrono::seconds(10)));

Action::Result action_ret = action->arm();
ASSERT_EQ(Action::Result::Success, action_ret);

telemetry->subscribe_flight_mode([&follow_me](Telemetry::FlightMode flight_mode) {
    const FollowMe::TargetLocation last_location = follow_me->get_last_location();

    std::cout << "[FlightMode: " << flight_mode
        << "]" Vehicle is at Lat: " << last_location.latitude_deg << " deg, "

```

```

        << "Lon: " << last_location.longitude_deg << " deg." << '\n';
    });

    action_ret = action->takeoff();
    ASSERT_EQ(Action::Result::Success, action_ret);

    sleep_for(seconds(5));

    // configure follow me behaviour
    FollowMe::Config config;
    config.follow_height_m = 12.f; // increase min height
    config.follow_distance_m = 20.f; // set distance b/w system and target during FollowMe
mode
    config.responsiveness = 0.2f; // Make it less responsive (higher value for the setting)
    config.follow_angle_deg = 0.0; // System follows target from FRONT side

    // Apply configuration
    FollowMe::Result config_result = follow_me->set_config(config);
    ASSERT_EQ(FollowMe::Result::Success, config_result);

    // Start following
    FollowMe::Result follow_me_result = follow_me->start();
    ASSERT_EQ(FollowMe::Result::Success, follow_me_result);

    // send location update every second
    send_location_updates(follow_me);

    // Stop following
    follow_me_result = follow_me->stop();
    ASSERT_EQ(FollowMe::Result::Success, follow_me_result);
    sleep_for(seconds(2)); // to watch flight mode change from "FollowMe" to default
"HOLD"

    action_ret = action->land();

```

```

ASSERT_EQ(Action::Result::Success, action_ret);
sleep_for(seconds(2)); // let it land

while (telemetry->armed()) {
    std::cout << "waiting for system to disarm" << "\n";
    sleep_for(seconds(1));
}
}

void print(const FollowMe::Config& config)
{
    std::cout << "Current FollowMe configuration of the system" << "\n";
    std::cout << "-----" << "\n";
    std::cout << "Height: " << config.follow_height_m << "m" << "\n";
    std::cout << "Distance: " << config.follow_distance_m << "m" << "\n";
    std::cout << "Following angle: " << config.follow_angle_deg << "[deg]" << "\n";
    std::cout << "Responsiveness: " << config.responsiveness << "\n";
    std::cout << "-----" << "\n";
}

FollowMe::TargetLocation    create_target_location(double    latitude_deg,    double
longitude_deg)
{
    FollowMe::TargetLocation location;
    location.latitude_deg = latitude_deg;
    location.longitude_deg = longitude_deg;
    location.absolute_altitude_m = 0.0;
    location.velocity_x_m_s = 0.f;
    location.velocity_y_m_s = 0.f;
    location.velocity_z_m_s = 0.f;
    return location;
}

bool autopilot_has_improved_followme(const std::shared_ptr<Param> param)

```

```

{
    // Check if the newly added parameter in the improved follow-me exists
    // Improved Follow-Me PR: https://github.com/PX4/PX4-Autopilot/pull/18026
    const std::pair<Param::Result, float> get_result = param-
>get_param_float("FLW_TGT_MAX_VEL");

    if (get_result.first == Param::Result::Success) {
        return true;
    } else {
        return false;
    }
}

void send_location_updates(std::shared_ptr<FollowMe> follow_me, size_t count, float rate)
{
    // TODO: Generate these co-ordinates from an algorithm
    // Altitude here is ignored by PX4, as we've set min altitude in configuration.
    std::array<FollowMe::TargetLocation, N_LOCATIONS> spiral_path = {
        create_target_location(47.39768399, 8.54564155),
        create_target_location(47.39776569, 8.54553292)};

    // We're limiting to N_LOCATIONS for testing.
    count = (count > N_LOCATIONS) ? N_LOCATIONS : count;
    std::cout << "# of Target locations: " << count << " @ Rate: " << rate << "Hz." << "\n";

    for (auto pos : spiral_path) {
        if (count-- == 0) {
            return;
        }
        follow_me->set_target_location(pos);
        auto sleep_duration_ms = static_cast<int>(1 / rate * 1000);
        sleep_for(milliseconds(sleep_duration_ms));
    }
}

```