

UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM  
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Desenvolvimento de um sistema portátil e de baixo custo de estação de  
controle em terra para VANTs

Douglas Aparecido Soares

Itajubá, junho de 2017

UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM  
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Douglas Aparecido Soares

Desenvolvimento de um sistema portátil e de baixo custo de estação de  
controle em terra para VANTs

Dissertação submetida ao Programa de Pós-Graduação em Ciência e  
Tecnologia da Computação como parte dos requisitos para obtenção  
do Título de Mestre em Ciência e Tecnologia da Computação

**Área de Concentração:** Matemática da Computação

**Orientador:** Prof. Dr. Alexandre Carlos Brandão Ramos

**Coorientador:** Prof. Dr. Roberto Affonso da Costa Junior

Junho de 2017

Itajubá-MG

À minha esposa Priscila, aos meus irmãos e demais familiares por todo o apoio e  
pelo tempo que deixamos de estar juntos...  
Aos meus pais Regina e João: a eles todos os créditos...

**Dedico**

## Agradecimentos

Agradeço...

Em primeiro lugar, a Deus, à Nossa Senhora, a São José e ao meu Santo Anjo da Guarda pelas bênçãos, graças e pela proteção constante.

À minha esposa, à minha família e à família da minha esposa por serem minha base e meu refúgio em todos os momentos.

Ao meu orientador, Prof. Dr. Alexandre Carlos Brandão Ramos, e ao meu coorientador, Prof. Dr. Roberto Affonso da Costa Junior, pela dedicação e apoio aos meus estudos neste período de aprendizado.

À UNIFEI, por ser um centro de excelência de ensino, que me permitiu cursar tanto a graduação em Engenharia da Computação quanto este curso de pós-graduação.

À CAPES, pela bolsa de mestrado no Programa de Pós-graduação em Ciência e Tecnologia da Computação, a qual serviu como apoio financeiro para a realização deste projeto.

À equipe Black Bee pela sua colaboração imprescindível para a minha dissertação.

Aos professores do POSCOMP por todo o conhecimento transmitido.

Aos meus amigos pela companhia, apoio e orações.

Aos meus colegas de pós-graduação, que tornaram um período de longa dedicação em algo divertido e de enorme aprendizado.

*“E Maria disse: Minha alma glorifica ao Senhor, meu espírito exulta de alegria em Deus, meu Salvador, porque olhou para sua pobre serva. Por isto, desde agora, me proclamam bem-aventurada todas as gerações, porque realizou em mim maravilhas aquele que é poderoso e cujo nome é Santo. Sua misericórdia se estende, de geração em geração, sobre os que o temem. Manifestou o poder do seu braço: desconcertou os corações dos soberbos. Derrubou do trono os poderosos e exaltou os humildes. Saciou de bens os indigentes e despediu de mãos vazias os ricos. Acolheu a Israel, seu servo, lembrado da sua misericórdia, conforme prometera a nossos pais, em favor de Abraão e sua posteridade, para sempre”*

Lc 1,46-55

## Resumo

Este trabalho visa o estudo minucioso do sistema *Paparazzi UAV* para o acompanhamento e controle de missões visando favorecer o desenvolvimento de futuros ambientes de *software* e *hardware* portáteis e de baixo custo para controle em terra de MiniVANTs (aeronaves de até 5Kg). O projeto utiliza o sistema operacional *Android* instalado em um *tablet*, permitindo ao usuário toda a mobilidade, a ergonomia e a facilidade para serviços de campo, características desse tipo de equipamento. O sistema preza pela portabilidade, uma vez que os seus componentes são, de modo geral, equipamentos mais leves e fáceis de transportar do que os tradicionalmente utilizados em uma estação de controle em terra. Além disso, os componentes e as ferramentas utilizadas no projeto são de baixo custo, aumentando a viabilidade de sua implementação. O controle de voo do VANT é feito através da comunicação entre o sistema *Paparazzi UAV*, instalado em uma placa *Raspberry Pi*, e o aplicativo *PPRZonDroid*, instalado no *tablet*. A estação de controle projetada possui uma vasta área de utilização, já que pode ser usada em várias aplicações, tanto militares quanto civis.

**Palavras-chave:** VANT, *Drone*, *Raspberry Pi*, *Paparazzi UAV*, *PPRZonDroid*, *Android*, Estação de Controle em Terra, GCS.

## Abstract

This project aims the rigorous study of the *Paparazzi UAV* system for the monitoring and control of missions with de objective of favoring the development of future portable, low-cost software and hardware environment to control the flight of a drone (aircraft up to 5Kg). The project uses the *Android* operating system of a tablet, what gives the user the benefits of the mobility, ergonomics and ease of field work that are characteristics of this device. The project focuses on portability. To achieve that, the chosen components are generally lighter and easier to transport rather than those traditionally used in a ground control station (GCS). In addition, the components and tools that were used in the project are low-cost, what increases the feasibility of its implementation. The RPA (Remotely Piloted Aircraft) flight control is performed through the communication between the *Paparazzi UAV* system and the *PPRZonDroid* application. It is convenient to say that the *Paparazzi UAV* and the *PPRZonDroid* application must be previously installed in a *Raspberry Pi* board and in a tablet, respectively. The projected ground control station has a wide area of utilization, since it can be used in various scenarios, both military and civilian.

**Key words:** RPA, UAV, Drone, *Raspberry Pi*, *Paparazzi UAV*, *PPRZonDroid*, *Android*, Ground Control Station, GCS.

## Sumário

<b>Lista de Figuras</b> .....	<b>9</b>
<b>Lista de Tabelas</b> .....	<b>11</b>
<b>Glossário</b> .....	<b>12</b>
<b>1 INTRODUÇÃO</b> .....	<b>13</b>
1.1 MOTIVAÇÃO.....	13
1.2 OBJETIVOS .....	17
1.3 ORGANIZAÇÃO DO TRABALHO .....	18
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>19</b>
2.1 ESTAÇÃO DE CONTROLE EM TERRA (GCS).....	19
2.1.1 Controle da Aeronave .....	19
2.1.2 Planejamento de Missões .....	20
2.1.3 Análise de Dados do Voo.....	20
2.2 SISTEMAS AÉREOS NÃO TRIPULADOS.....	21
2.3 PAPERAZZI UAV .....	22
2.4 PPRZonDroid .....	24
2.5 RASPBERRY PI.....	25
2.6 REDES DE COMUNICAÇÃO.....	27
2.6.1 Wi-Fi.....	27
2.6.2 Protocolo ZigBee.....	27
2.7 ANDROID.....	29
<b>3 SISTEMA PROJETADO</b> .....	<b>31</b>
3.1 SOFTWARE .....	31
3.1.1 Central Paparazzi.....	32
3.1.1.1 Configuração.....	33
3.1.1.2 Compilação .....	33
3.1.1.3 Execução .....	33
3.1.2 Estação de Controle em Terra .....	34
3.2 HARDWARE .....	36
3.2.1 Autopiloto e Placa Microcontroladora.....	36
3.2.2 Unidade de Medida Inercial.....	38
3.2.3 Sistema de Posicionamento Global (GPS).....	41
3.2.4 Modems de Comunicação.....	42

3.2.5	Rádio, Receptor RC e Conversor de Sinais .....	43
3.3	INTEGRAÇÃO SOFTWARE-HARDWARE .....	46
<b>4</b>	<b>METODOLOGIA .....</b>	<b>50</b>
<b>5</b>	<b>RESULTADOS E DISCUSSÃO .....</b>	<b>58</b>
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>60</b>
	Referências .....	61



## Lista de Figuras

Figura 1 – Primeiro exemplo de VANT.....	13
Figura 2 – Segundo exemplo de VANT.....	14
Figura 3 – Terceiro exemplo de VANT.....	14
Figura 4 – Visão geral do projeto <i>Paparazzi UAV</i> .....	22
Figura 5 – Tela principal do <i>PPRZonDroid</i> .....	24
Figura 6 – Placa de desenvolvimento <i>Raspberry Pi</i> Modelo B+ V1.2.....	26
Figura 7 – Módulo <i>Xbee-XSC SMA</i> .....	29
Figura 8 – Arquitetura da Plataforma <i>Android</i> .....	29
Figura 9 – <i>Central Paparazzi</i> .....	32
Figura 10 – Exemplo de uma aeronave realizando um voo autônomo mostrado na GCS.....	35
Figura 11 – Vista inferior da placa <i>Lisa/M V2.0</i> .....	37
Figura 12 – Vista superior da placa <i>Lisa/M V2.0</i> , detalhando a sua pinagem.....	38
Figura 13 – Vista superior da placa <i>Aspirin IMU</i> , detalhando a sua pinagem.....	39
Figura 14 – Vista superior da placa <i>Aspirin IMU</i> , destacando o seu pequeno tamanho.....	40
Figura 15 – Vista inferior da placa <i>Aspirin IMU</i> , destacando o seu pequeno tamanho.....	40
Figura 16 – <i>GPS U-blox LEA 5S</i> .....	41
Figura 17 – <i>GPS U-blox LEA 5S</i> conectado no circuito principal.....	42
Figura 18 – <i>Modem XBee Pro ZB</i> com saída <i>pigtail</i> .....	43
Figura 19 – <i>Modem XBee Pro ZB</i> com antena pequena embutida.....	43
Figura 20 – Rádio com chave de três posições.....	44
Figura 21 – Receptor do rádio.....	44
Figura 22 – Placa <i>ATmega PPM Encoder</i> .....	45
Figura 23 – Conexões dos dispositivos que vão embarcados na aeronave.....	46
Figura 24 – Placa FTDI e <i>modem</i> de comunicação que são conectados à placa <i>Raspberry Pi</i> .....	47
Figura 25 – Resumo da metodologia empregada.....	50
Figura 26 – Esquema da comunicação entre o <i>PPRZonDroid</i> e o VANT.....	52
Figura 27 – <i>Tablet Google HTC Nexus 9</i> executando o aplicativo <i>PPRZonDroid</i> .....	53
Figura 28 – VANT (quadrotor) utilizado no projeto.....	55
Figura 29 – Esquema mostrando alguns dos componentes da GCS desenvolvida..	56

Figura 30 – Fluxograma com a ideia básica deste projeto.....	57
Figura 31 – Suporte para facilitar o manuseio de um <i>tablet</i> e um RC.....	57

## Lista de Tabelas

Tabela 1 – Principais especificações do <i>Tablet Google HTC Nexus 9</i> .....	54
--	----

## Glossário

**AHRS** – em português: Altitude e posição do sistema de referência

**ANAC** – Agência Nacional de Aviação Civil

**ARP** – Aeronave Remotamente Pilotada

**CAPES** – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

**GCS** – *Ground Control Station*

**GPS** – *Global Positioning System*

**INS** – em português: Sistema de navegação integrado

**IMU** – *Inertial Measurement Unit*

**ITA** – Instituto Tecnológico da Aeronáutica

**ITNG** – *International Conference on Information Technology: New Generations*

**POSCOMP** – Programa de Pós-Graduação em Ciência e Tecnologia da Computação

**PPM** – *Pulse-Position Modulation*

**RF** – Rádio Frequência

**SO** – Sistema Operacional

**UAV** – *Unmanned Aerial Vehicle*

**UNIFEI** – Universidade Federal de Itajubá

**VANT** – Veículo Aéreo Não Tripulado

# 1 INTRODUÇÃO

Neste capítulo inicial será feita uma breve introdução, além de algumas definições essenciais relacionadas ao tema abordado neste trabalho: os veículos aéreos não tripulados.

## 1.1 MOTIVAÇÃO

Veículo aéreo não tripulado (VANT) ou Aeronave Remotamente Pilotada (ARP) é todo e qualquer tipo de aeronave que não necessita de pilotos embarcados para ser guiada. A título de curiosidade, o termo “drone” (em português: zumbido) é apenas uma denominação genérica, originada nos EUA, porém sem amparo técnico ou definição na legislação brasileira. Por outro lado, o termo correto para definir este tipo de aeronave de acordo com a legislação brasileira é VANT [1]. Por este motivo, nessa dissertação será utilizado o termo VANT.

De um modo geral, os VANTs podem ser categorizados de várias formas, por exemplo: quanto ao tamanho, à altitude máxima alcançada, ao tipo de combustível utilizado, etc. Nas figuras a seguir (1 a 3) são mostrados alguns tipos diferentes de VANTs de modo a exemplificar a grande variedade de aeronaves que vêm sendo desenvolvidas.



Figura 1 – Primeiro exemplo de VANT

Fonte: Disponível em: <<http://www.decea.gov.br/static/uploads/2015/01/vant-2000.jpg>>. Data do acesso (Nov, 14), 2016



Figura 2 – Segundo exemplo de VANT

Fonte: Disponível em: <<http://geodrones.com.br/wp-content/uploads/2015/08/drone-vant-EVENT-38-STD-img2.jpg>>. Data do acesso (Nov, 14), 2016



Figura 3 – Terceiro exemplo de VANT

Fonte: Disponível em: <[http://mundogeo.com/wp-content/uploads/2013/09/EnsoMOSAIC\\_UAV\\_Mini.jpg](http://mundogeo.com/wp-content/uploads/2013/09/EnsoMOSAIC_UAV_Mini.jpg)>. Data do acesso (Nov, 14), 2016

O avanço de diversas tecnologias relacionadas a veículos aéreos não tripulados fizeram com que o seu potencial para uso civil fosse elevado, considerando que a maior parcela de uso, atualmente, encontra-se em fins militares [2]. Como os VANTs são sistemas versáteis, as possibilidades de sua utilização variam incluindo o monitoramento de linhas de transmissão [3], o apoio à segurança pública [4] e à agricultura de precisão [5]. Além disso, podem ser utilizados para outras diversas aplicações, dentre as quais podem ser citadas: a obtenção de informações (fotos, vídeos, etc.) para realizar a vigilância em tempo real de uma certa região [6]; a pesquisa em determinadas áreas geográficas de interesse, verificando, por exemplo, as modificações na vegetação ou nos recursos hídricos, bem como o efeito que as variações climáticas provocam em um determinado local [7]; o monitoramento do

avanço de tropas e no lançamento de bombas e mísseis [8]. Com o passar dos anos têm sido realizadas cada vez mais pesquisas nesta área e tem ocorrido a criação de diversas empresas dedicadas ao desenvolvimento de UAVs (do Inglês, *Unmanned Aerial Vehicles*) e equipamentos e tecnologias visando o aperfeiçoamento dos VANTs.

No Brasil, dentre as principais aplicações para as quais os VANTs são utilizados, podem ser citadas, especialmente, o mapeamento aéreo e terrestre, o controle de pragas nas plantações [9, 10], a inspeção de linhas de transmissão de energia elétrica [11], a vigilância de uma determinada área, o controle de recursos hídricos e vegetações [12], etc. Além dessas aplicações, tem-se observado cada vez mais pesquisas sobre VANTs no País, incluindo diversos temas como, por exemplo, o processamento de imagens obtidas por VANTs [13] usando esses dados para as mais diversas finalidades.

A ideia deste trabalho é desenvolver uma estação de controle em terra portátil e de baixo custo para realizar o controle de VANTs. A portabilidade é uma característica desejada pensando em facilitar o transporte da estação de controle, bem como viabilizar que a mesma seja utilizada em diversos ambientes. O baixo custo aumenta a possibilidade de que este sistema seja implantado e utilizado em diversas áreas uma vez que a estação de controle em terra desenvolvida é versátil, podendo ser utilizada tanto em atividades civis quanto militares.

Além disso, o foco deste trabalho é utilizar o sistema desenvolvido para o controle de MiniVANTs. Esta escolha foi feita, principalmente, pelos seguintes motivos:

- Redução de custos: os MiniVANTs são mais baratos do que as outras categorias de VANTs;
- Praticidade/portabilidade: o manuseio e transporte de MiniVANTs facilita a operação em campo já que se tratam de aeronaves mais leves do que outros tipos de VANTs;
- Este tipo de VANT tem grande demanda na sociedade, sendo utilizado em diversas aplicações.

No sistema desenvolvido são utilizados uma placa de desenvolvimento *Raspberry Pi* e um *tablet* no lugar de um computador (ou *notebook*) que seria tradicionalmente utilizado em uma estação de controle em terra. Além disso, o sistema

de piloto automático utilizado (*Paparazzi UAV*) é de código aberto, o que flexibiliza a utilização do projeto, já que o uso de ferramentas de código aberto aumenta a possibilidade de personalização na operação das mesmas, além de permitirem eventuais modificações que sejam necessárias em seu código fonte. De um modo geral, *softwares* comerciais (que possuem código fechado) apresentam menor possibilidade de personalização, além de não possibilitarem acesso ao seu código fonte.



## 1.2 OBJETIVOS

Este trabalho visa o desenvolvimento de um ambiente de *software* e *hardware* portátil e de baixo custo que permita realizar o controle em terra de MiniVANTs (aeronaves de até 5 Kg). A estação de controle em terra (em Inglês, *Ground Control Station - GCS*) projetada tem o intuito de facilitar a interação com os VANTs quando operados em campo, viabilizando o planejamento e a execução de missões do sistema, além de permitir o acompanhamento dos dados de voo de forma simultânea. O projeto utiliza o sistema operacional *Android* instalado em um *tablet*, permitindo ao usuário toda a mobilidade, a ergonomia e a facilidade para serviços de campo, características desse tipo de equipamento. O sistema projetado preza pela portabilidade, uma vez que os seus componentes são, de modo geral, equipamentos mais leves e fáceis de transportar do que os tradicionalmente utilizados em uma estação de controle em terra. Além disso, os componentes e as ferramentas utilizadas no projeto são de baixo custo, aumentando a viabilidade de sua implantação.

### 1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho foi organizado de forma a facilitar o entendimento do tema abordado. Vale ressaltar que haverá vários termos citados em Inglês, pois possuem uma definição mais concisa na Língua Inglesa e, dentre eles, há alguns que nem mesmo possuem tradução para o Português, sendo comumente utilizados os mesmos termos que no Inglês.

No Capítulo 2 é apresentada a fundamentação teórica, contendo uma breve explanação sobre diversos temas e conceitos essenciais para a compreensão deste trabalho.

No Capítulo 3 é feita uma descrição sobre o sistema projetado, abordando os componentes de *software* e *hardware* utilizados, bem como a forma como eles foram integrados durante a execução do trabalho.

A metodologia utilizada durante o desenvolvimento do projeto é descrita no Capítulo 4.

No Capítulo 5 são apresentados os resultados do desenvolvimento deste projeto, além de serem discutidas várias soluções para os mais variados problemas encontrados durante a construção do sistema projetado.

Finalmente, no Capítulo 6 são apresentadas as conclusões do trabalho, além de possíveis trabalhos futuros relacionados ao tema abordado, o qual possui uma vasta área de aplicação.

## 2 FUNDAMENTAÇÃO TEÓRICA

Para o entendimento do funcionamento do sistema desenvolvido, ao longo deste capítulo será realizada uma breve abordagem sobre alguns tópicos e conceitos de fundamental importância para este trabalho.

### 2.1 ESTAÇÃO DE CONTROLE EM TERRA (GCS)

Uma estação de controle em terra, terminal em terra ou central de controle em terra é uma estação terrestre composta por um sistema de rádios, além de equipamentos eletrônicos e computacionais destinada a realizar a comunicação com o VANT, enviando e recebendo sinais e dados, realizando, assim, o controle e o monitoramento do voo da aeronave [14].

A estação de controle é um dos mais importantes componentes de um sistema de VANT. O núcleo de uma GCS é formado por um computador, sendo que o *software* da GCS é considerado seu elemento mais crítico [15], pois a boa operação do sistema de controle depende das características e eficiência do *software*. Os principais objetivos de uma GCS são: controle da aeronave, planejamento de missões autônomas e visualização e processamento de dados durante o voo.

#### 2.1.1 Controle da Aeronave

O controle pode ser feito de forma direta, alterando diretamente a orientação da aeronave através do uso, por exemplo, de um controle remoto, ou de um modo indireto, através de modificações no plano de voo que esteja sendo realizado de forma autônoma pelo VANT.

### 2.1.2 Planejamento de Missões

Missões são planos de voo, definidos pelo operador, que são realizados de forma autônoma pelos VANTs. Uma forma de definir uma missão é através da definição de *waypoints*, que são pontos tridimensionais que devem ser atingidos pela aeronave durante o voo.

Dependendo do *software* do piloto automático utilizado podem estar disponíveis diversos tipos de *waypoints*, dentre os quais podem ser citados *waypoints* pontuais, de decolagem, de pouso, de espera e de retorno à casa (corresponde ao ponto de origem do voo definido pelo operador).

### 2.1.3 Análise de Dados do Voo

É de extrema importância na operação de VANTs a habilidade de perceber e monitorar o ambiente nas proximidades da aeronave, tomando as ações necessárias para evitar acidentes ou mesmo contornar erros durante a execução do plano de voo que foi previamente definido. O monitoramento dos dados do voo é necessário para que o operador do sistema na GCS tenha uma melhor percepção da situação do voo, uma vez que não há piloto embarcado na aeronave. Dentre os principais dados que podem ser monitorados durante o voo podem ser citados: a altitude atual, a velocidade do ar, a velocidade de solo, a orientação espacial, a posição espacial, a localização geográfica, o modo de controle/voo, a altitude e a velocidade desejados, a situação da missão, o consumo de energia, etc. [16]

## 2.2 SISTEMAS AÉREOS NÃO TRIPULADOS

Um sistema aéreo não tripulado compreende uma série de subsistemas que incluem a aeronave, sua carga, a estação de controle e subsistemas de comunicação, dentre outros. [17]

Conforme citado anteriormente, os veículos aéreos não tripulados são definidos como aeronaves que não transportam uma tripulação, podendo ser controlados remotamente ou mesmo voar de maneira autônoma. Neste trabalho serão abordados unicamente os sistemas aéreos não tripulados para aeronaves que realizam voos autônomos.

Um dos principais sistemas aéreos não tripulados é o *ArduPilot* (informações mais detalhadas sobre o *ArduPilot* podem ser encontradas em [18]) que é um controlador de voo de código aberto que foi construído baseado na plataforma *Arduino* (mais informações sobre o *Arduino* podem ser encontradas em [19]) e que possui a capacidade de realizar voos autônomos permitindo estabilização autônoma, navegação por *waypoints* e obtenção de dados de telemetria [20]. Em [20] e em [21] são apresentados trabalhos nos quais o *ArduPilot* foi utilizado como o sistema controlador de voo de um VANT.

O sistema aéreo não tripulado usado neste trabalho foi o *Paparazzi UAV*. A definição inicial dos componentes de *software* e *hardware* que seriam utilizados neste projeto (início de 2015) foi feita com base nos equipamentos que estavam disponíveis na UNIFEI para a pesquisa relacionada a VANTs. Mais especificamente, a placa controladora de voo que estava disponível possuía o sistema *Paparazzi* embarcado. Além disso, no início do desenvolvimento deste trabalho, sistemas similares ao que foi projetado já estavam sendo desenvolvidos utilizando a tecnologia *ArduPilot* e, portanto, a utilização do *Paparazzi* seria de maior contribuição à área acadêmica e para a comunidade como um todo.

## 2.3 PAPERAZZI UAV

O *Paparazzi UAV* é um projeto de código aberto criado em 2003 composto por um sistema de *software* e *hardware*, abrangendo o sistema de autopilotagem e o *software* para controle em terra de multicópteros e multirotores, aeronaves de asa fixa, helicópteros e aeronaves híbridas. Seu foco principal é o voo autônomo e tem como foco secundário o voo manual. Possui um sistema dinâmico de plano de voo, que é definido pelos estados da missão e o uso de pontos de passagem (*waypoints*) como variáveis. Deste modo, é possível criar facilmente missões complexas totalmente automatizadas, sem a intervenção de operadores humanos. [22]

A utilização do *Paparazzi* foi adequada para este projeto por ser um sistema mais voltado à pesquisa e por contar com uma comunidade ativa de usuários e desenvolvedores, a qual serviu de grande apoio durante o desenvolvimento do projeto.

Na figura 4 é mostrada a visão geral do projeto *Paparazzi* com o esquema básico de comunicação entre os componentes de uma GCS.

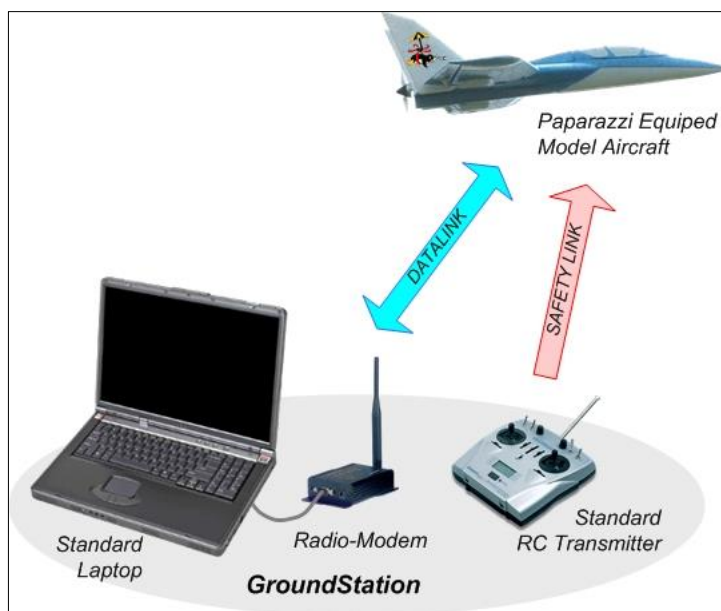


Figura 4 – Visão geral do projeto *Paparazzi UAV*

Fonte: Disponível em: <[https://wiki.paparazziuav.org/wiki/File:Paparazzi\\_System\\_overview.jpg](https://wiki.paparazziuav.org/wiki/File:Paparazzi_System_overview.jpg)>. Data do acesso (Nov, 16), 2016

No esquema mostrado na figura 4 é demonstrada a operação de uma GCS tradicional, na qual o *notebook* (*standard laptop*) e o conjunto rádio-modem permitirão

a comunicação com a aeronave através de um link de dados (*data link*). De modo a garantir a segurança na operação do VANT é ainda disponibilizado um link de segurança (*safety link*) no qual um operador humano toma o controle do VANT através da utilização de um rádio controle (*standard RC transmitter*).

Neste trabalho, o *notebook* é substituído por uma placa de desenvolvimento *Raspberry Pi* (mais informações na seção 2.5) trazendo maior portabilidade para a estação de controle do VANT. O *software* do sistema *Paparazzi UAV* foi instalado na placa *Raspberry Pi* e o *hardware* do *Paparazzi* foi embarcado no VANT.

## 2.4 PPRZonDroid

*PPRZonDroid* é um aplicativo utilizado para controlar um VANT através de um dispositivo móvel com o sistema operacional *Android*. O servidor da aplicação e o aplicativo *Android* são conectados através de uma rede *Wi-Fi*, de modo que eles estejam na mesma rede. Vários clientes podem ser conectados ao mesmo tempo, com acesso total ou restrito. Esse aplicativo, que ainda está em fase de desenvolvimento, teoricamente funciona no *Android* 3.2 ou superior. [23]

No sistema desenvolvido neste trabalho, o *PPRZonDroid* foi instalado num *tablet Google HTC Nexus 9* (mais detalhes sobre o *tablet* serão dados no capítulo 4). O *PPRZonDroid* se comunica com o sistema *Paparazzi UAV* que está instalado na placa *Raspberry Pi*. No início do desenvolvimento deste trabalho (início de 2015) não foram encontradas referências em outros trabalhos em que era realizada a comunicação entre o *PPRZonDroid* e o *Paparazzi UAV*, com o *Paparazzi UAV* sendo instalado numa *Raspberry Pi*.

Na figura 5 é mostrado um exemplo da tela principal do aplicativo *PPRZonDroid* na qual é apresentada uma simulação de voo de dois VANTs (trajetórias azul e vermelha).

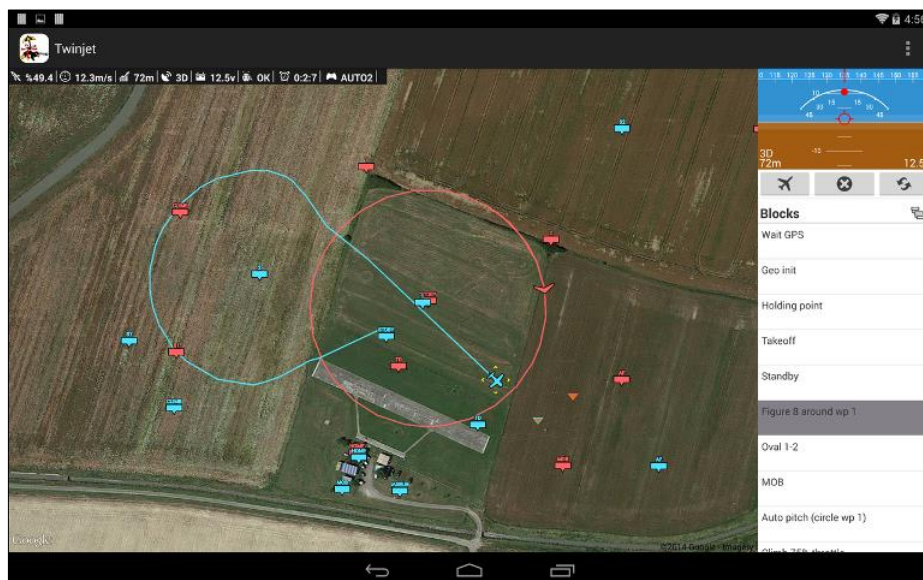


Figura 5 – Tela principal do *PPRZonDroid*

Fonte: Disponível em: <[https://wiki.paparazziuav.org/wiki/File:PPRZonDroid\\_tablet.png](https://wiki.paparazziuav.org/wiki/File:PPRZonDroid_tablet.png)>. Data do acesso (Nov, 16), 2016



## 2.5 RASPBERRY PI

*Raspberry Pi* é um minicomputador criado pela *Raspberry Pi Foundation* com o objetivo de estimular o ensino da Ciência da Computação nas escolas e universidades. Apesar de o *Raspberry Pi* possuir o *hardware* em uma única placa eletrônica de tamanho reduzido, seu potencial de processamento é significativo, permitindo a sua utilização em diversos projetos tecnológicos. [24]

A *Raspberry Pi* proporciona uma boa experiência para os seus usuários, pois pode ser programada em diversas linguagens de melhor compreensão como *Java*, *C* e *Python*. Além disso, por possuir um custo relativamente baixo (a placa em suas versões mais básicas custa cerca de US\$ 30,00) permite o desenvolvimento de diversas aplicações com um baixo custo, característica de grande relevância para este trabalho. [25]

A placa *Raspberry Pi* pode ser encontrada em dois modelos (A e B) e está, atualmente, na terceira fase do seu desenvolvimento, sendo nesta fase denominada *Raspberry Pi 3*. As placas de modelo B apresentam melhores características computacionais do que as de modelo A, porém apresentam custo ligeiramente maior. Apesar disso, ambos os modelos possuem o mesmo tamanho e são compatíveis com sistemas operacionais baseados em *Linux*. Foi desenvolvida uma distribuição *Linux* oficial para o *Raspberry Pi*, a qual é chamada *Raspbian*. Entretanto, outras distribuições do *Linux* como *Debian*, *Ubuntu* e *Arch Linux* também são compatíveis com a placa. O processador da placa *Raspberry Pi* é um ARM. [26]

Neste trabalho foi utilizada uma placa *Raspberry 1 Modelo B+*, a qual foi lançada em 2014. O módulo funciona com um processador *Broadcom BCM2835* de núcleo único e 700 MHz, o mesmo presente em todos os *Raspberry Pi* lançados até novembro de 2014, e conta com 512 MB de memória RAM de 400 MHz no antigo padrão DDR2. [27]

Para ligar periféricos e *hardware* diversos, a placa conta com interface GPIO de 40 pinos. Há quatro portas USB 2.0, saída HDMI, *slot* para cartão de memória do tipo microSD, baixo consumo de energia (entre 0,5 e 1 watt) e saída de som P2. O modelo também tem *Ethernet*, interface para câmera e tela. [27]

A figura 6 mostra uma visão geral da placa *Raspberry Pi* usada neste projeto.

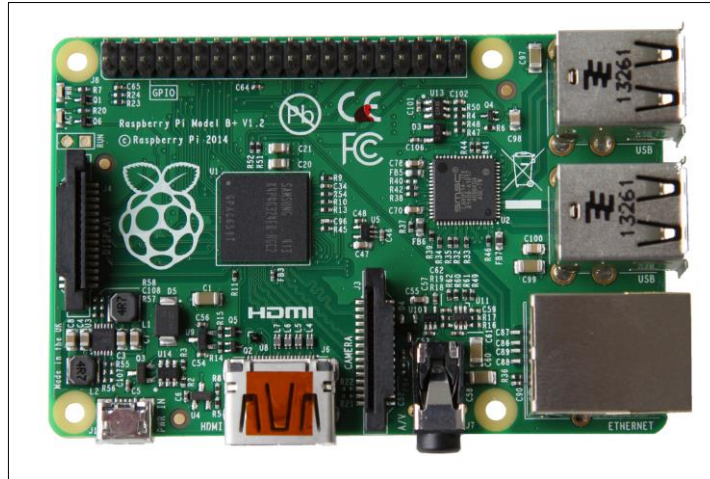


Figura 6 – Placa de desenvolvimento *Raspberry Pi* Modelo B+ V1.2

Fonte: Disponível em: <[https://www.raspberrypi.org/wp-content/uploads/2014/07/rsz\\_b-.jpg](https://www.raspberrypi.org/wp-content/uploads/2014/07/rsz_b-.jpg)>. Data do acesso (Nov, 16), 2016

## 2.6 REDES DE COMUNICAÇÃO

A seguir serão descritas as principais tecnologias e protocolos de comunicação utilizados neste projeto para viabilizar a troca de informações entre a GCS e o VANT.

### 2.6.1 Wi-Fi

No contexto dos VANTs as redes sem fio são utilizadas para criar o *link* de telemetria da aeronave, para que seja possível observar e controlar remotamente os dados do veículo em operação. Dentre as limitações observadas no uso de redes *wireless* em VANTs, podem ser citadas, especialmente, a baixa potência disponível, limitações de espaço físico e as distâncias relativamente grandes a serem superadas. [28]

Neste trabalho, foi utilizada uma rede *Wi-Fi* para permitir a comunicação entre o sistema *Paparazzi UAV* (instalado na *Raspberry Pi*) e o aplicativo *PPRZonDroid* (instalado no *tablet*).

### 2.6.2 Protocolo ZigBee

O protocolo *ZigBee* permite a comunicação sem fio entre dispositivos eletrônicos através da utilização de sinais de radiofrequência. Apresenta baixa potência de operação, baixa taxa de transmissão de dados e baixo custo de implantação. A tecnologia utilizada é comparável às redes *Wi-Fi* e *Bluetooth* e diferencia-se destas por empregar menor consumo de energia [29], o que é interessante no contexto da operação de VANTs.

A economia de energia por parte dos módulos RF (Rádio Frequência) que utilizam o padrão *ZigBee* é viabilizada pelo fato de que quando não estão transmitindo ou recebendo dados, os módulos *ZigBee* entram num estado de dormência (em Inglês, *sleep mode*), consumindo menos energia. [30]

Na figura 7 é mostrado o módulo *XBee* utilizado neste projeto, que se trata do Módulo *Xbee-XSC SMA*. A integração e a utilização deste módulo neste trabalho serão descritas no capítulo 3.



Figura 7 – Módulo *Xbee-XSC SMA*

Fonte: Disponível em: <<https://images.digi.com/products/xbee-xsc-rpsma>>. Data do acesso (Mai, 11), 2017

## 2.7 ANDROID

*Android* é um sistema operacional desenvolvido pela *Open Handset Alliance* destinado a dispositivos móveis e que foi construído tendo como base o núcleo do *Linux*. O sistema teve grande popularização desde seu ano de lançamento público, em 2008, o que ocorreu devido à combinação de sua facilidade de uso com o fato de sua disponibilidade ser gratuita. Juntamente com o próprio sistema, observou-se o crescimento da utilização dos aplicativos desenvolvidos para este SO (Sistema Operacional). De modo geral, o desenvolvimento de aplicativos se dá através de programação utilizando a linguagem de programação *Java*, que é uma linguagem de programação orientada à objetos e que possui uma sintaxe similar às linguagens *C/C++*. [31]

Na figura 8 é apresentada a arquitetura da plataforma *Android*, detalhando os seus principais componentes.



Figura 8 – Arquitetura da Plataforma *Android*

Fonte: Disponível em: <<https://image.slidesharecdn.com/plataformaandroidpt-130206183517-phpapp01/95/arquitetura-da-plataforma-android-1-638.jpg?cb=1367228745>>. Data do acesso (Mai, 03), 2017

A opção pelo uso do *Android* neste trabalho deu-se por algumas características que são oferecidas por este sistema operacional e que são relevantes para este projeto. Dentre elas, podem ser citadas: é um sistema de código aberto que permite o uso de novas tecnologias e tem acesso aos recursos implementados no *kernel* do *Linux*; é adotado por uma grande variedade de dispositivos no mercado, além de ser compatível com dispositivos de várias marcas; garante ao usuário liberdade de customização; possui uma enorme quantidade de aplicativos já implementados e há cada vez mais pesquisadores e desenvolvedores nesta área [32]. Todas essas características do *Android* aumentam a gama de utilização do projeto implementado.

### 3 SISTEMA PROJETADO

Um dos principais objetivos deste trabalho é a realização de um estudo minucioso da plataforma *Paparazzi UAV* visando o desenvolvimento de futuras aplicações para uma estação de controle em terra para VANTs, com o foco em pequenos veículos aéreos. Normalmente estes veículos são operados em linha de visão, com pouco tempo para configurar uma base de comando. Dentre as características e funcionalidades especificadas para o sistema projetado podem ser citados, principalmente:

- Portabilidade: facilitaria a utilização da GCS em trabalhos de campo;
- Visualização de dados de voo (telemetria, condições de voo, velocidade e posicionamento do VANT, entre outros) em tempo real;
- Planejamento de missões autônomas;
- Controle do VANT;
- Possibilidade de configurar parâmetros do voo do VANT como, por exemplo, a trajetória a ser seguida de forma autônoma, a velocidade da aeronave, etc.;
- Baixo peso da aeronave: este projeto visa o controle de MiniVANTs (aeronaves de até 5 Kg);
- Baixo custo do projeto: de modo que aumente a viabilidade da implementação deste projeto.

Neste capítulo será feita uma descrição sobre o sistema projetado, detalhando a integração do *software* e do *hardware* utilizados na construção e desenvolvimento deste trabalho.

#### 3.1 SOFTWARE

Na primeira parte deste capítulo serão detalhados os principais componentes de *software* utilizados no projeto, que são a *Central Paparazzi* e a Estação de Controle em Terra (GCS).

### 3.1.1 Central Paparazzi

A *Central Paparazzi* (em Inglês, *Paparazzi Center*) é uma interface gráfica que possui três páginas principais [33]:

- Um conjunto de caixas de seleção para configurar uma aeronave com o seu plano de voo e elaborar os programas a serem simulados e enviados para o veículo aéreo;
- Um painel de controle para inicializar os Agentes de Execução do sistema (*link* de dados, servidor da aplicação, GCS e receptor/transmissor de mensagens trocadas entre a GCS e o VANT) e manipular uma coleção de programas através de sessões configuráveis;
- Uma página onde a GCS pode ser incorporada.

Além destas principais telas, um Console de log (mostra os dados trocados entre a GCS e o VANT, além de status da execução do plano de voo configurado) também é exibido nas páginas de configuração e do painel de controle [33]. A figura 9 mostra um exemplo de tela da *Central Paparazzi*.

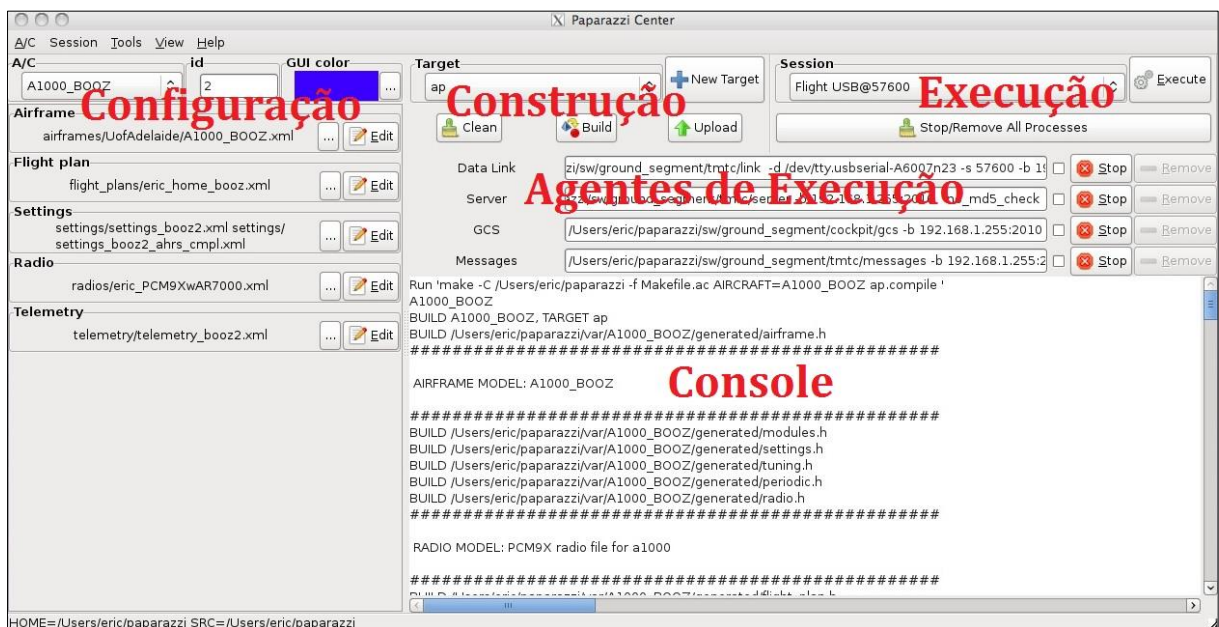


Figura 9 – *Central Paparazzi*

Fonte: Disponível em: <<http://blog.paparazziuav.org/wp-content/uploads/2010/11/paparazzicenter.jpg>>. Data do acesso (Mai, 09), 2017



A seguir serão descritas as principais etapas para a utilização da *Central Paparazzi*, as quais compreendem a: Configuração, Construção (Compilação) e Execução (destacados na figura 9).

#### 3.1.1.1 Configuração

No painel de Configuração é possível definir as configurações da aeronave e o seu plano de voo. As configurações feitas no editor ficam salvas no arquivo *conf.xml*, o qual pode ser editado tanto através da própria interface gráfica da *Central Paparazzi* quanto manualmente acessando o arquivo de configuração que fica salvo na pasta aonde o sistema *Paparazzi UAV* foi instalado. A partir do menu *A/C* (figura 9) é possível tanto acrescentar uma nova aeronave quanto excluir a aeronave atual. [33]

#### 3.1.1.2 Compilação

A compilação é feita a partir do painel de Construção. Usando o botão novo alvo (em Inglês, *New Target*, que pode ser visualizado na figura 9) pode-se adicionar novos alvos para a compilação. Os comandos relacionados à compilação são mostrados como Agentes de Execução para que eles possam ser interrompidos, se necessário. [33]

É possível construir programas com rotinas que serão executadas durante o voo do VANT em várias linguagens de programação de alto nível como C, Java e MATLAB [33]. Neste trabalho todos os programas foram construídos utilizando a linguagem C.

#### 3.1.1.3 Execução

No painel de Execução é possível selecionar a seção desejada a partir de um conjunto de sessões pré-definidas. A sessão de simulação compreende um servidor, uma GCS e um simulador para a aeronave que foi selecionada no painel de

Configuração. É importante ressaltar que o alvo da simulação (*sim*) deve ter sido construído antes de a simulação ser realizada. [33]

A partir das opções de configuração da execução é possível parar e reiniciar a execução dos programas, editar configurações previamente definidas, além de salvar os conjuntos dos processos atuais como uma sessão para que possa ser reiniciada em outro momento. [33]

Após o usuário ter clicado no botão *Execute*, que pode ser visualizado na figura 9, (em Português, executar) é aberta a interface gráfica da GCS na qual é possível visualizar os dados tanto de uma simulação quanto de um plano de voo que foi projetado na *Central Paparazzi* e que esteja sendo executado pelo VANT. [33]

### 3.1.2 Estação de Controle em Terra

A Estação de Controle em Terra (GCS) é um *software* utilizado pelo operador para realizar o controle de veículos aéreos. Ele permite visualizar e controlar um veículo aéreo durante o desenvolvimento e a operação, tanto em ambiente *indoor* quanto *outdoor*. Possui uma arquitetura de *software* que suporta vários tipos de aeronaves, além de projetos de piloto automático. O objetivo da GCS é realizar o monitoramento em tempo real de um VANT. [34]

A GCS é uma aplicação que possui interfaces gráficas personalizáveis, com componentes que possuem utilidade para os mais variados propósitos. Dentre as suas características e funcionalidades podem ser citadas [34]:

- Suporte para o voo simultâneo de várias aeronaves;
- Suporte multi-sistemas (múltiplos protocolos, pilotos automáticos e projetos);
- Mapa 2D capaz de exibir imagens do *Google Satellite*, *OpenStreetMaps* e *Microsoft Satellite Maps*;
- Planejamento de missões de voo;
- Mudanças de *waypoints* em tempo real;
- Ajustes dos planos de voo em tempo real, se necessário;
- Visão geral do status do sistema;
- Calibração da aeronave em tempo real, mesmo estando no ar;

- Suporta aeronaves de asas rotativas e de asas fixas como, por exemplo, aviões, helicópteros coaxiais e quadrotores;
- Teclas de atalho configuráveis para acesso rápido em trabalhos de campo;
- Interface configurável.

Na figura 10 é mostrado um exemplo de tela da GCS na qual pode ser vista uma aeronave realizando um voo autônomo com trajetória circular que foi previamente projetado na *Central Paparazzi*.

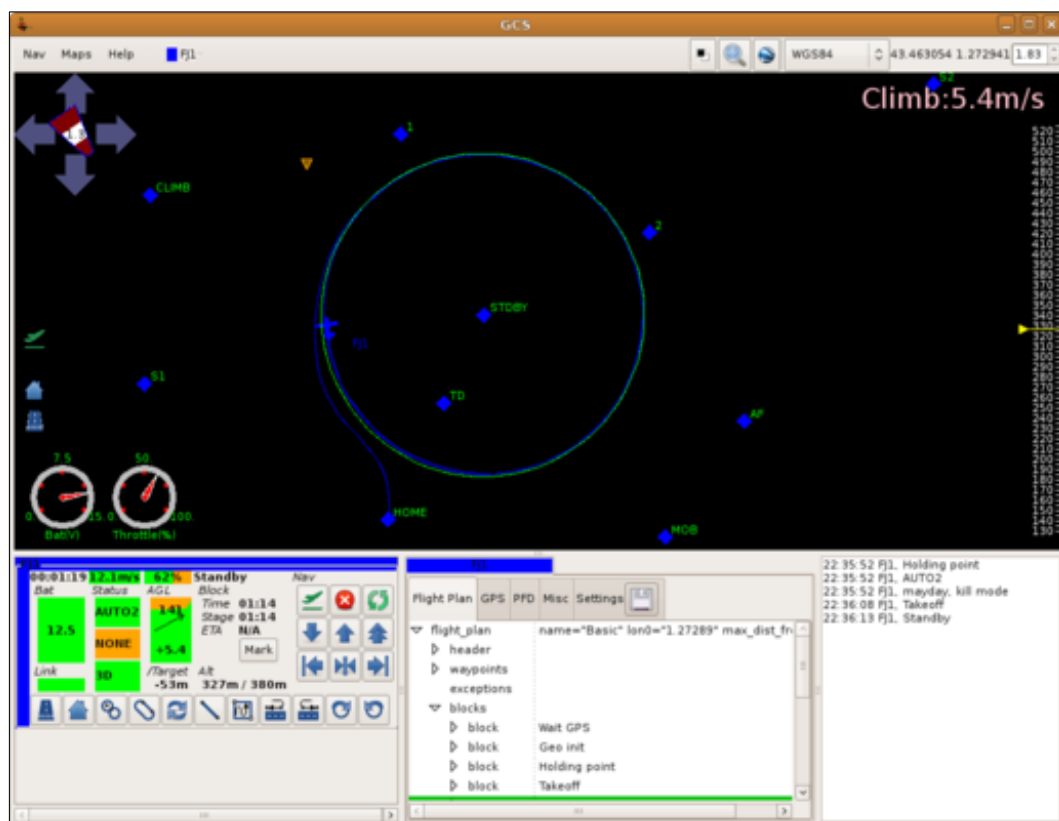


Figura 10 – Exemplo de uma aeronave realizando um voo autônomo mostrado na GCS

Fonte: Disponível em: <<https://wiki.paparazziuav.org/w/images/thumb/a/ae/Papgets.png/516px-Papgets.png>>. Data do acesso (Mai, 09), 2017

## 3.2 HARDWARE

Nesta seção serão detalhados os principais componentes de *hardware* utilizados no projeto, abordando suas principais funcionalidades e características.

### 3.2.1 Autopiloto e Placa Microcontroladora

Neste projeto o sistema escolhido como autopiloto do VANT foi o sistema de código aberto (em Inglês, *open source*) *Paparazzi UAV*. Uma característica importante do *Paparazzi* é que ele possui suporte para projetos que utilizem diversos tipos de *hardware*, implicando em boas possibilidades de implementação. [35]

Além disso, para o desenvolvimento deste trabalho a placa microcontroladora escolhida foi a placa *Lisa/M*, a qual possui diversas características que são relevantes para este projeto.

*Lisa/M* é uma placa miniatura de piloto automático, de uso geral, e que possui flexibilidade para uso em múltiplas plataformas. Este microcontrolador foi concebido a partir de melhorias na placa microcontroladora *STM32*. Possui um processador rápido, tem pequeno peso e tamanho e pode possuir integrada a ela a placa *Aspirin IMU*. Devido ao seu pequeno tamanho e as suas especificações, o uso da *Lisa/M* é adequado para os veículos de asa fixa e de asa rotativa. [36]

Como mencionado acima, a placa *Lisa/M* é baseada no processador de 64 pinos *STM32F105RCT6*. Possui 64 KB de memória RAM e 256 KB de memória *flash*. Todos os seus pinos são expostos (disponíveis para conexões), permitindo acesso ao conjunto completo de periféricos *STM32*. Dentre as suas principais especificações podem ser citadas [36]:

- Microcontrolador *STM32F105RCT6* com 256 KB de memória *flash* e 64 KB de memória RAM;
- Sensor de pressão *BMP085*;
- 7 (sete) canais de entrada analógicos;
- 3 (três) canais de entrada/saída digitais genéricos;
- 2 (dois) 3.3V TTL UART (5V);
- 8 (oito) saídas servo PPM;

- 1 (um) barramento CAN;
- 1 (um) barramento SPI;
- 1 (um) barramento I2C;
- 1 (um) micro USB;
- 4 (quatro) LEDs de status com teste de ponto integrado;
- Tem 10,8 gramas (com a *Aspirin IMU* montada);
- Tem 9,9 gramas (sem a *Aspirin IMU* montada);
- Dimensões de 34 mm x 60 mm x 10 mm;
- Placa de circuito impresso (PCB) de 4 camadas.

Na figura 11 é apresentada uma imagem com a vista inferior da placa *Lisa/M V2.0* (versão da placa utilizada neste projeto). Nesta imagem é possível observar o processador da placa *Lisa/M*, além de outros circuitos integrados auxiliares e da descrição dos pinos e conexões.

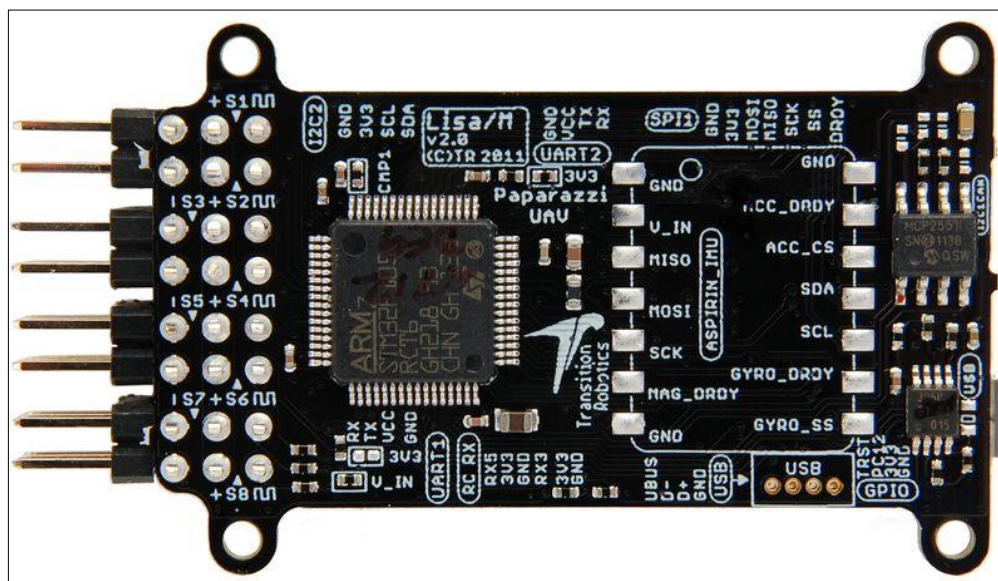


Figura 11 – Vista inferior da placa *Lisa/M V2.0*

Fonte: Disponível em: <[http://wiki.paparazziuav.org/wiki/File:LisaM\\_V2\\_0\\_BottomView.JPG](http://wiki.paparazziuav.org/wiki/File:LisaM_V2_0_BottomView.JPG)>. Data do acesso (Mai, 11), 2017

Já a figura 12 é uma representação da vista superior da placa *Lisa/M V2.0* com seus respectivos pinos de conexão (maiores detalhes sobre a piagem da placa *Lisa/M V2.0* podem ser encontrados em [37]) para a montagem elétrica do projeto.

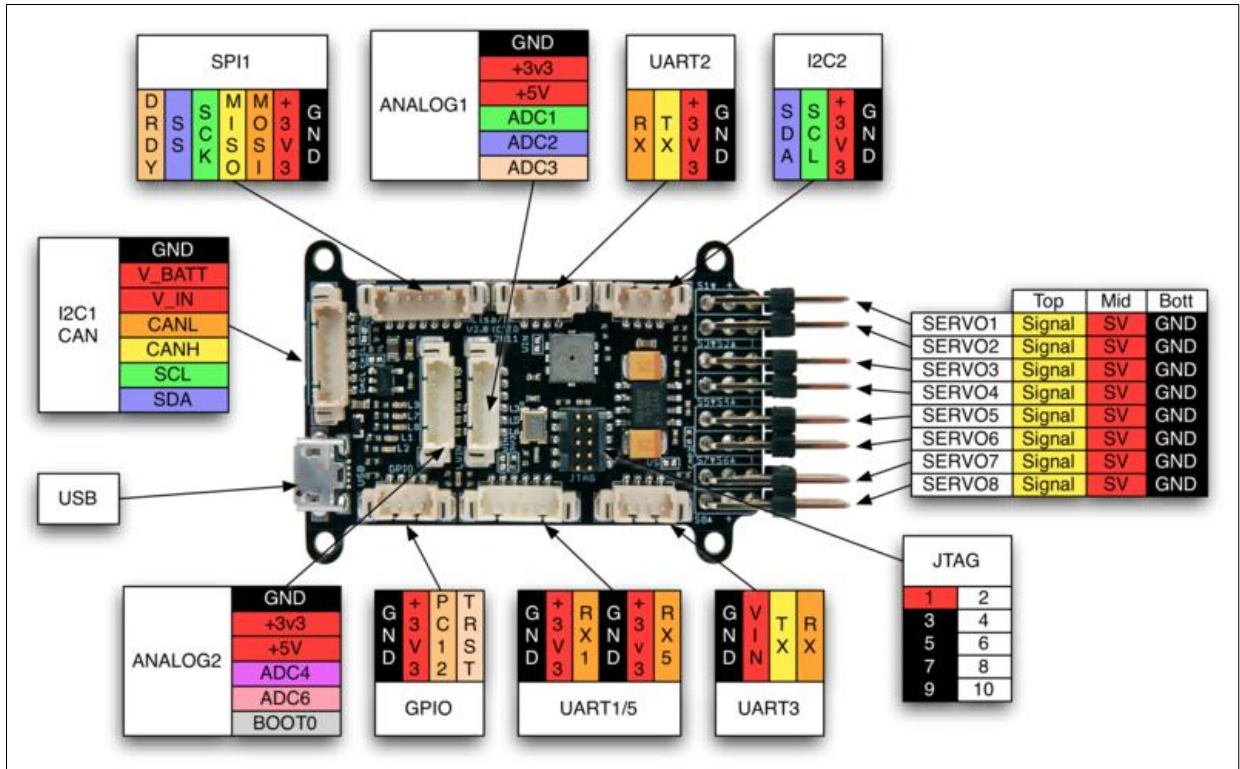


Figura 12 – Vista superior da placa *Lisa/M V2.0*, detalhando a sua pinagem

Fonte: Disponível em: <[http://wiki.paparazziuav.org/wiki/File:LisaM\\_V2\\_0\\_top\\_labeled.png](http://wiki.paparazziuav.org/wiki/File:LisaM_V2_0_top_labeled.png)>. Data do acesso (Mai, 11), 2017

### 3.2.2 Unidade de Medida Inercial

A Unidade de Medida Inercial (em Inglês, *Inertial Measurement Unit – IMU*) é formada por um conjunto de 3 sensores tri-axiais: um acelerômetro, um giroscópio e um magnetômetro. De um modo geral, a unidade formada por esses sensores é capaz de medir qualquer movimento de um corpo no mesmo sistema inercial. IMUs são amplamente utilizadas para estabilização, determinação de posicionamento e controle de voo. Além disso, a IMU pode ser usada para medir as acelerações, rotações angulares e, em alguns casos, o campo magnético da Terra. [38]

Neste trabalho foi utilizada a *Aspirin IMU V2.1* uma vez que esta IMU possui altas compatibilidade e sincronia com a placa microcontroladora escolhida, conforme mencionado anteriormente, a placa *Lisa/M V2.0*. A *Aspirin IMU* usa sensores digitais de 3 eixos e possui um tamanho reduzido e baixo custo. Outras características incluem reguladores de tensão *on-board*, EEPROMs para armazenamento de dados de

calibração e sensores extras, como por exemplo, um barômetro. Dentre as principais características da *Aspirin IMU* podem ser citadas [39]:

- Acelerômetro de 3 eixos;
- Giroscópio de 3 eixos;
- Magnetômetro de 3 eixos;
- Regulagem de voltagem *on-board*;
- EEPROM I2C *on-board*;
- Barômetro *on-board*;
- *Footprint* padrão entre todas as versões;
- PCB *flat form-factor* único;
- Encaixe e pinagem compatível para a placa *Lisa/M*.

A figura 13 mostra detalhes sobre os pinos da placa IMU utilizada neste projeto. Já as figuras 14 e 15 mostram as vistas superior e inferior desta placa, destacando o seu pequeno tamanho.

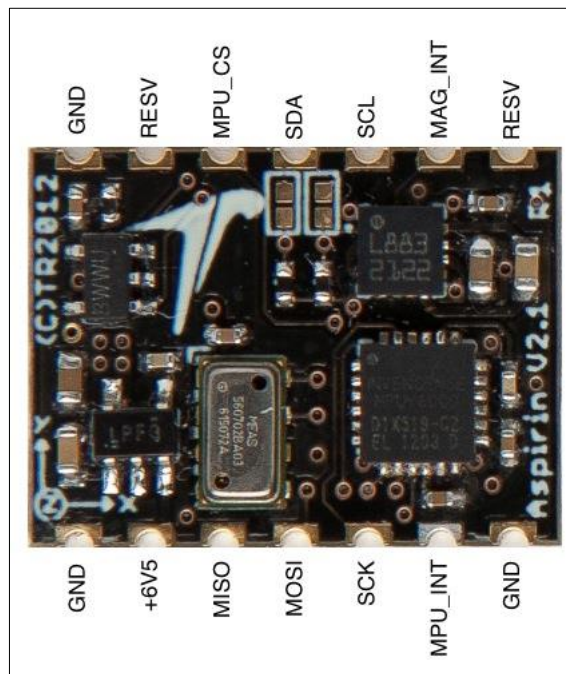


Figura 13 – Vista superior da placa *Aspirin IMU*, detalhando a sua pinagem

Fonte: Disponível em: <[https://wiki.paparazziuav.org/wiki/File:Aspirin\\_IMU\\_documented.jpg](https://wiki.paparazziuav.org/wiki/File:Aspirin_IMU_documented.jpg)>. Data do acesso (Mai, 11), 2017



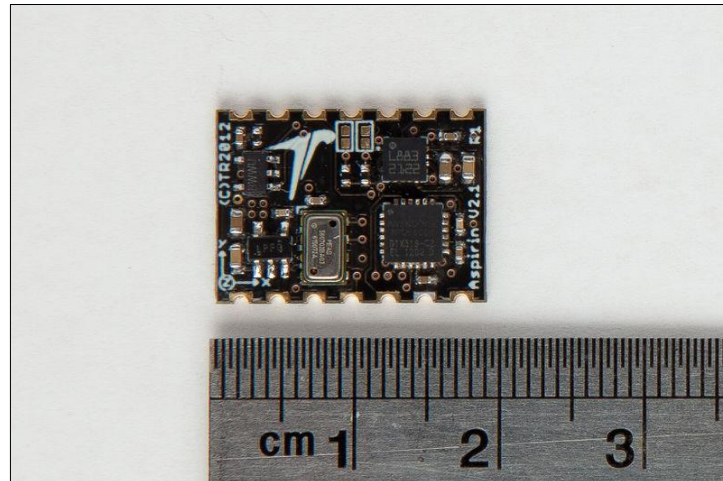


Figura 14 – Vista superior da placa *Aspirin IMU*, destacando o seu pequeno tamanho

Fonte: Disponível em: <[https://wiki.paparazziuav.org/wiki/File:Aspirin\\_2.1\\_top.jpg](https://wiki.paparazziuav.org/wiki/File:Aspirin_2.1_top.jpg)>. Data do acesso (Mai, 11), 2017

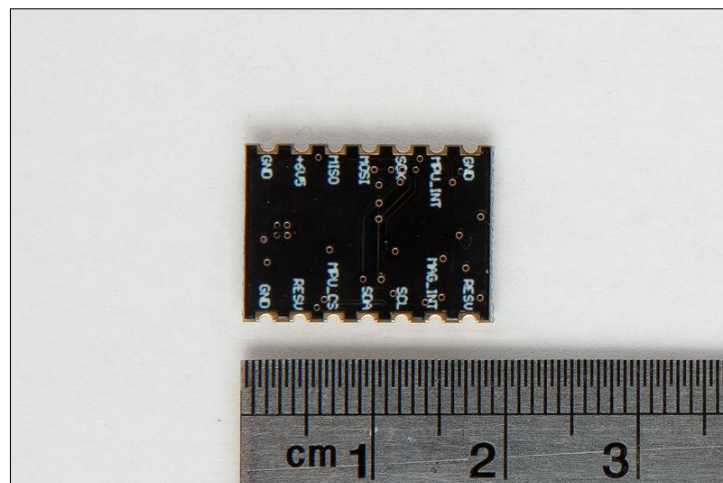


Figura 15 – Vista inferior da placa *Aspirin IMU*, destacando o seu pequeno tamanho

Fonte: Disponível em: <[https://wiki.paparazziuav.org/wiki/File:Aspirin\\_2.1\\_bottom.jpg](https://wiki.paparazziuav.org/wiki/File:Aspirin_2.1_bottom.jpg)>. Data do acesso (Mai, 11), 2017

A IMU em conjunto com outros dispositivos fornece dados que viabilizam a execução de algumas funções no sistema. São elas:

- *AHRS*: altitude e posição do sistema de referência. Utiliza dados da IMU e de outros dispositivos (por exemplo: GPS, sensor de velocidade do ar, barômetro) para fazer a fusão de sensores e fornecer os movimentos de arfagem (em Inglês, *pitch* – refere-se ao movimento de inclinação de um avião) e giro da aeronave; [38]



- *INS*: sistema de navegação integrado. Usa a IMU e outros sensores de navegação (por exemplo: GPS) além de algoritmos complexos para fornecer a posição e a velocidade do VANT. [38]

### 3.2.3 Sistema de Posicionamento Global (GPS)

Um receptor de GPS (sistema de geoposicionamento global – em Inglês, global positioning system) é necessário para o posicionamento da aeronave em campos abertos (voos *outdoor*) e para programar rotas e missões nas quais se precisa saber a posição da aeronave. [40]

Neste trabalho foi empregado um receptor *U-Blox LEA 5S*, uma vez que este receptor de GPS apresenta boa compatibilidade com o sistema *Paparazzi UAV*. [41]

Na figura 16 é apresentada uma imagem do GPS utilizado neste projeto. Já na figura 17 é mostrada uma imagem da montagem do circuito principal no qual o receptor de GPS é integrado.



Figura 16 – GPS *U-blox LEA 5S*

Fonte: Disponível em: <<https://sc02.alicdn.com/kf/HTB1JXmbHFXXXXcwXXXXq6xXFXXN/Hot-offer-positioning-engine-lea-5s-ublox.jpg>>. Data do acesso (Mai, 11), 2017



Figura 17 – GPS U-blox LEA 5S conectado no circuito principal

Fonte: Autor

#### 3.2.4 Modems de Comunicação

Neste trabalho foram necessários dois *modems* para viabilizar a comunicação entre a estação de controle em terra e o aeromodelo. Um *link* bidirecional entre estes dois *modems* permite que as informações de telemetria cheguem à tela da GCS em tempo real. Além de dados de telemetria, outras funções são possíveis a partir desta comunicação como, por exemplo, o ajuste fino de variáveis na aeronave, assim como comandos de navegação e mudanças de pontos de passagem (*waypoints*) durante o voo.

Os *modems* utilizados são de modelo *XBee Pro ZB*, os quais se comunicam através da utilização do protocolo *ZigBee* (descrito na seção 2.6.2). Um dos *modems* tem uma saída *pigtail* para que o usuário possa conectar nesta saída a antena de sua preferência. Um destes *modems* fica inserido na GCS juntamente com um sistema para fazer a comunicação com uma das portas USB da placa *Raspberry Pi*. O segundo *modem* vai embarcado no VANT e possui uma pequena antena embutida de modo que o seu tamanho não seja empecilho para ser usado em qualquer que seja o tipo de aeronave empregada.

Nas figuras a seguir é possível visualizar o primeiro *modem* com a saída *pigtail* (figura 18) e o segundo *modem* com a sua própria antena embutida (figura 19).

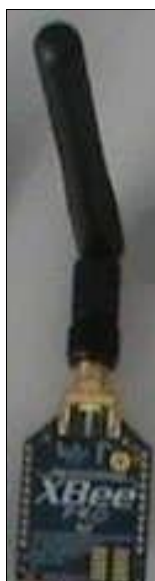


Figura 18 – *Modem XBee Pro ZB com saída pigtail*

Fonte: Autor



Figura 19 – *Modem XBee Pro ZB com antena pequena embutida*

Fonte: Autor

### 3.2.5 Rádio, Receptor RC e Conversor de Sinais

Para que seja possível realizar todos os testes no *software* e no *hardware* usados neste projeto, tentando minimizar a possibilidade de acidentes na operação do VANT, deve-se ter a possibilidade de trocar entre os comandos automáticos,

semiautomáticos e totalmente manuais. A possibilidade de trocar para os comandos manuais, mesmo este projeto lidando com veículos aéreos que realizam voos autônomos, é necessária para que, caso haja alguma falha no sistema, um piloto experiente possa assumir totalmente o controle da aeronave trocando a posição da chave de seleção do modo de comando no rádio. Para tanto, um rádio com uma chave de três posições se fez necessário.

Nas figuras a seguir são mostrados o rádio (figura 20) e o receptor (figura 21) utilizados neste trabalho.



Figura 20 – Rádio com chave de três posições

Fonte: Autor



Figura 21 – Receptor do rádio

Fonte: Autor

Além do receptor e do rádio também foi necessário o uso de uma placa transformadora (decodificadora) de sinais, a qual recebe o sinal conhecido como modulação por posição de pulso (em Inglês, *Pulse-Position Modulation - PPM*). Este sinal transforma os vários sinais recebidos pelo receptor em um único sinal de saída necessário para que o *software* que está embarcado no VANT entenda os comandos a serem executados.

A figura 22 mostra o decodificador PPM utilizado neste trabalho, o decodificador *ATmega PPM Encoder*.



Figura 22 – Placa *ATmega PPM Encoder*

Fonte: Autor

### 3.3 INTEGRAÇÃO SOFTWARE-HARDWARE

Nesta seção serão feitos comentários sobre como foi feita a integração do *software* e do *hardware* [42] utilizados neste trabalho.

Em um primeiro momento, para a montagem do *hardware* que vai embarcado no VANT, foi necessário soldar a IMU ao fundo da placa de autopiloto *Lisa/M*. Os dispositivos restantes foram todos ligados à placa de piloto automático através de conectores *molex picoblade*, exceto os servos da aeronave e a placa *PPM encoder*, que foram ligadas através de conectores servos, os quais são amplamente empregados no aeromodelismo tradicional.

Na saída denominada UART3 da placa *Lisa/M* foi conectado o módulo de GPS; o *modem* de comunicação foi ligado à saída UART2 da *Lisa/M*; e o servo 6 da *Lisa/M* foi conectado à *PPM encoder*. Na *PPM encoder* foram conectados todos os sinais do receptor de rádio controle, completando todos os dispositivos que serão embarcados na aeronave.

A figura 23 mostra a conexão entre os dispositivos descrita acima. A caneta presente na figura serve apenas como referência para demonstrar o tamanho dos dispositivos, demonstrando a viabilidade de que os mesmos sejam embarcados em um Micro UAV.

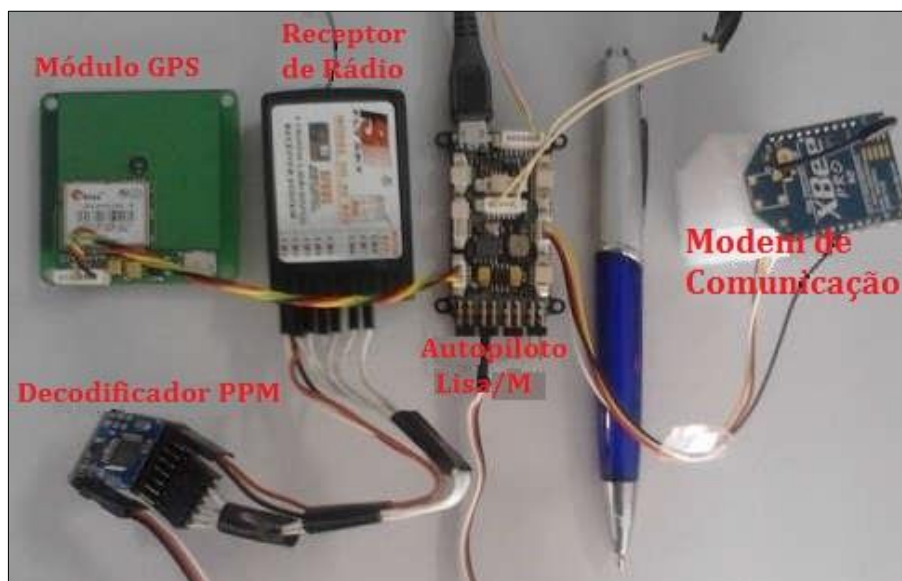


Figura 23 – Conexões dos dispositivos que vão embarcados na aeronave

Fonte: Autor

Outro dispositivo importante para o envio e a recepção de mensagens entre a estação de controle e a aeronave é o outro *modem*. Este *modem* foi ligado à placa *Raspberry Pi* e, para que essa conexão fosse possível, foi necessária uma placa intermediária chamada FTDI. O *modem* é conectado na placa FTDI, a qual possui uma saída USB para ser ligada a outros dispositivos, no caso à *Raspberry Pi*.

A figura 24 mostra o *modem* e a sua placa FTDI que foram conectadas à placa *Raspberry Pi*.



Figura 24 – Placa FTDI e *modem* de comunicação que são conectados à placa *Raspberry Pi*

Fonte: Autor

Já para a configuração do *software* utilizado neste trabalho, iniciou-se com a instalação da estação de controle em terra na placa *Raspberry Pi*. O sistema operacional que foi utilizado é a distribuição *Linux Ubuntu versão 12.04.5 LTS*. O código fonte do sistema *Paparazzi UAV* está hospedado no *Github* [43], e para fazer o *download* do repositório, usando a ferramenta de versionamento *Git* [44], o seguinte comando foi utilizado no *Terminal* do *Linux* (todos os comandos demonstrados a seguir devem ser executados no *Terminal*):

```
git clone --origin upstream https://github.com/paparazzi/paparazzi.git
```

A seguir, outro comando se fez necessário indicando a versão do sistema *Paparazzi* que será baixada. No caso do comando abaixo a versão baixada do sistema *Paparazzi* seria a 5.6:

***git checkout v5.6***

Para realizar a compilação do código fonte é preciso acessar o diretório no qual o *Paparazzi* foi baixado através do comando ***cd ~/paparazzi*** (considerando que o *Paparazzi* foi baixado no *home* do usuário do *Linux*) e, então, executar o comando ***run*** e, por fim, para iniciar o *software* usar o comando ***./paparazzi***.

A partir deste momento já é possível executar uma simulação na *Central Paparazzi*. Um exemplo de simulação que pode ser feito é escolhendo o tipo "Microjet" dentro de "aircraft" e selecionando "sim" como *target*, depois basta clicar em "Build" esperar alguns segundos até que seja finalizado o processo de construção da simulação e depois clicar em "Execute" com a sessão de "Simulation" selecionada. Caso o *software* tenha sido configurado corretamente e os passos anteriores tenham sido corretamente executados será possível ver o resultado da simulação na interface gráfica da *Central Paparazzi*.

Para executar o código em uma das aeronaves reais toma-se como padrão inicial o modelo "Bixler" encontrado nos arquivos do sistema *Paparazzi*. Após realizar as alterações (se necessárias) no modelo inicial, para executar o código e enviá-lo para a placa *Lisa/M* é necessário alterar o "Target" dentro do *Paparazzi Center* para "ap" (indicando uma aplicação real), clicar em "Build" e, depois, realizar o "Upload" de modo que a *Lisa/M* receba os arquivos configurados para a aeronave apropriada.

As principais alterações que devem ser feitas podem ser encontradas nas primeiras linhas do arquivo de configurações referente à aeronave. Tais linhas servem para selecionar se o modelo utilizado será uma aeronave de asa fixa ou uma aeronave de asas rotativas e também para identificar qual placa de piloto automático será utilizada. O código ficaria de forma semelhante à demonstrada a seguir:

```
<firmware name="fixedwing or rotorcraft"> <!-- Selecione o tipo de aeronave -->  
<target name="ap" board="lisa_m_2.0"/> <!-- Selecione a placa utilizada -->  
...  
</firmware>
```



Um detalhe importante a ser ressaltado é o fato de que, ao usar transmissores com sinal PPM, o que ocorre neste trabalho, conforme descrito na seção 3.2.5, o arquivo de rádio deve ser configurado com bastante cuidado, uma vez que erros de configuração neste arquivo podem gerar erros significativos no funcionamento do sistema como um todo, como, por exemplo, falhas no controle do VANT no caso de o operador realizar o controle manual do voo do VANT através do rádio controle. Neste arquivo é possível encontrar todos os canais do rádio com suas respectivas funções, inclusive a função de seleção de modo, pela qual é possível trocar entre comando manual, comando manual com estabilização automática e comando completamente autônomo.

## 4 METODOLOGIA

Neste capítulo serão abordadas diversas etapas e métodos utilizados para a elaboração e execução deste trabalho.

Na figura 25 é apresentado um resumo da metodologia empregada no desenvolvimento deste trabalho, a qual será explicada em detalhes neste capítulo.

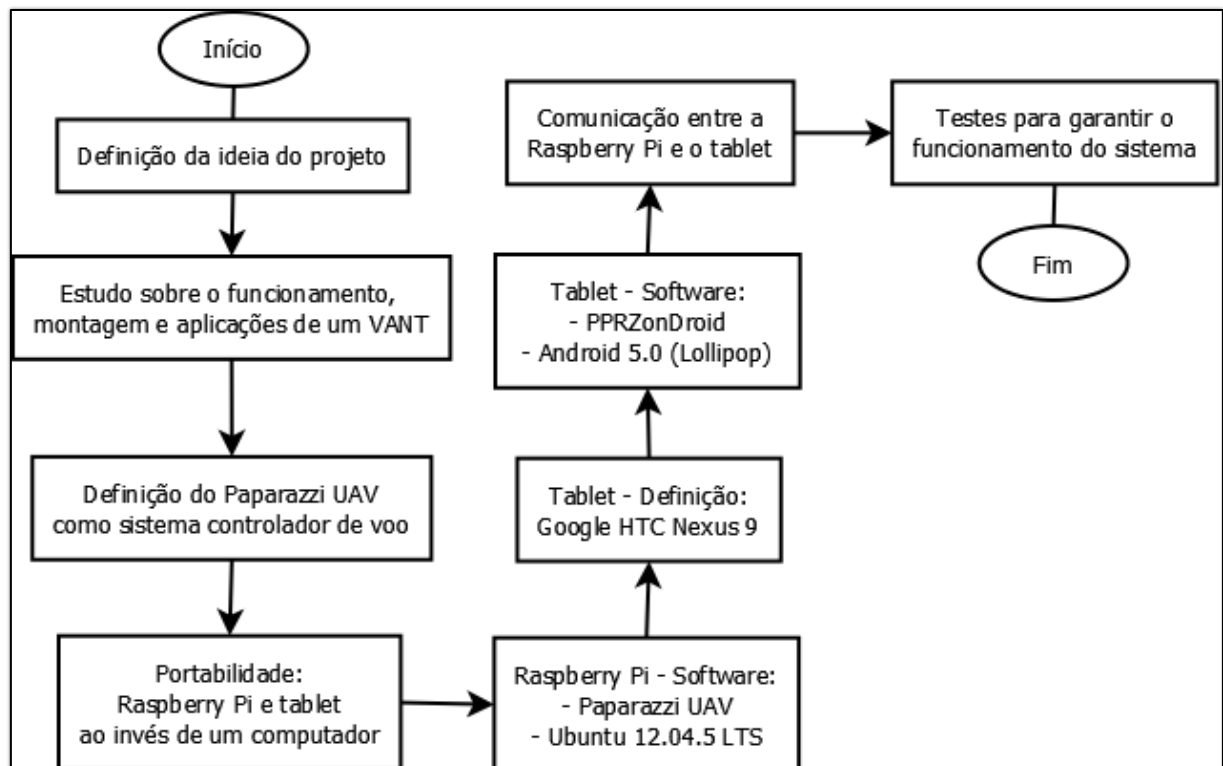


Figura 25 – Resumo da metodologia empregada

Fonte: Autor

Para o desenvolvimento deste projeto foi preciso passar por algumas etapas essenciais. Em um primeiro momento, foi realizado um estudo sobre o funcionamento de um VANT (alguns detalhes sobre o *hardware* e a montagem do VANT utilizado no projeto podem ser vistos no Capítulo 3), bem como as técnicas de seu controle de voo, além de algumas aplicações para as quais vem sendo utilizado. [45, 46, 47]

Para o controle de voo do VANT foram utilizados os *softwares Paparazzi UAV* (instalado em uma placa de desenvolvimento *Raspberry Pi*) e *PPRZonDroid* (instalado em um tablet com sistema operacional *Android*).

Uma etapa importante para o bom funcionamento do sistema projetado foi a definição de qual sistema operacional seria utilizado na *Raspberry Pi*. Dentre as várias opções testadas (diversas versões dos sistemas operacionais *Raspbian* e *Linux* foram utilizadas), o sistema operacional escolhido foi o *Ubuntu 12.04.5 LTS*. Essa escolha foi feita uma vez que esta versão do SO foi a única, dentre as testadas, que permitiu a plena instalação e utilização do *Paparazzi UAV*. Em todos os outros sistemas testados sempre houve algum erro ou durante a instalação ou durante o uso do *Paparazzi*. O processo de instalação do *Paparazzi UAV* em diversos sistemas operacionais pode ser encontrado em [48].

Após esta etapa foi feito um estudo para a definição sobre como se realizaria a comunicação entre o *Paparazzi UAV* (instalado na *Raspberry Pi*) e o *PPRZonDroid* (instalado em um *tablet Google HTC Nexus 9*). Esta etapa é essencial para o projeto, uma vez que, estabelecer a comunicação inicial e mantê-la segura e operacional, permite que sejam realizadas diversas atividades e sejam executadas várias aplicações a partir dos dados transferidos e recebidos pelo VANT durante o voo. Uma observação relevante acerca desse assunto é que o *Paparazzi UAV* e o *PPRZonDroid* precisam estar na mesma rede e para isso foi utilizada a rede *Wi-Fi* disponibilizada no campus da UNIFEI, em Itajubá-MG. O *tablet* conecta-se diretamente à rede através do seu próprio sistema de *software* e *hardware*. Já a *Raspberry Pi* foi conectada à rede *wireless* através da utilização de um adaptador *dongle Wi-Fi*.

Uma vez que o *Paparazzi UAV* e o *PPRZonDroid* estejam na mesma rede, para colocar este projeto em execução é preciso seguir alguns procedimentos: deve-se iniciar o servidor da aplicação no *Paparazzi UAV*; além disso, deve-se ou configurar um novo tipo de aeronave no *Paparazzi* ou então selecionar algum dos tipos básicos que o *software* oferece; após iniciar a simulação no *Paparazzi* deve-se executar o aplicativo *PPRZonDroid* no *tablet*. Para que a comunicação funcione é preciso que tanto o *Paparazzi* quanto o *PPRZonDroid* estejam configurados conforme descrito em [23]. A figura 26 mostra o esquema básico, descrito acima, para a comunicação entre o *PPRZonDroid* e um VANT equipado com o *Paparazzi UAV*.

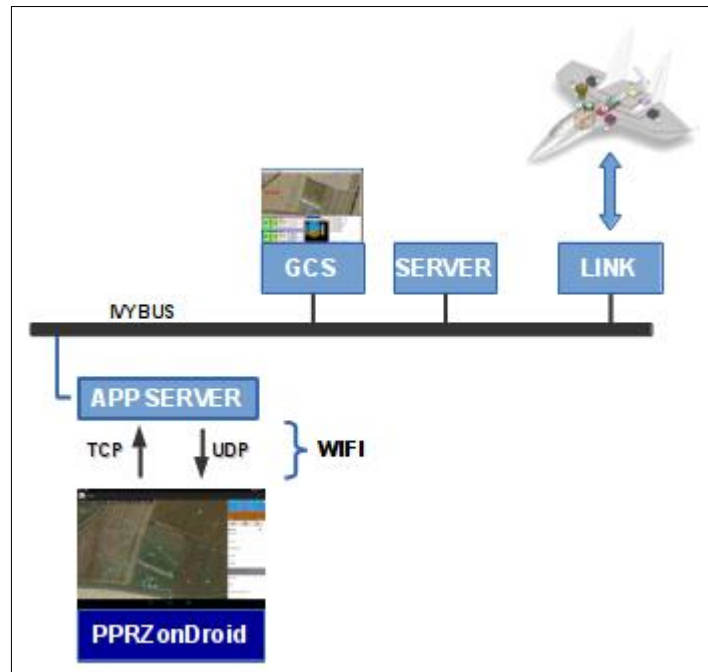


Figura 26 – Esquema da comunicação entre o *PPRZonDroid* e o VANT

Fonte: Disponível em: <<https://wiki.paparazziuav.org/wiki/File:PPRZonDroidOverview.png>>. Data do acesso (Mai, 12), 2017

A estação de controle é o equipamento utilizado em campo para o planejamento, controle e visualização do voo do VANT. Para atender a todas as especificações de projeto, e buscando reduzir o custo do sistema, optou-se pela utilização de um *tablet* com o sistema operacional *Android* instalado. A utilização do *Android* garante ao projeto as seguintes vantagens, dentre outras:

- O *Android* é um sistema de código aberto que permite o uso de novas tecnologias e tem acesso aos recursos implementados no *kernel* do *Linux*;
- É adotado por uma enorme variedade de dispositivos no mercado, além de ser compatível com dispositivos de várias marcas;
- Garante ao usuário liberdade de customização;
- Possui uma vasta gama de aplicativos já implementados para este SO e há cada vez mais pesquisadores e desenvolvedores nesta área.

Todas essas características do *Android* aumentam a gama de utilização do projeto implementado.

Para a escolha do *tablet* a ser utilizado no projeto foram consideradas dentre outras características, a capacidade de armazenamento e processamento e a versão

do sistema operacional *Android* a ser utilizada. Como a única restrição deste projeto quanto a estação de controle é de que o *hardware* seja compatível com o sistema operacional *Android* a gama de dispositivos disponíveis no mercado é grande.

Selecionou-se o tablet *Nexus 9* da empresa *Google* (em parceria com a empresa *HTC*) por atender os requisitos desta aplicação, além do fato de ser considerado um dispositivo de referência para o sistema *Android*, em virtude de sua boa eficiência computacional.

Na figura 27 é apresentada a tela inicial do aplicativo *PPRZonDroid* sendo executado no *tablet Nexus 9*, que foi utilizado neste projeto.

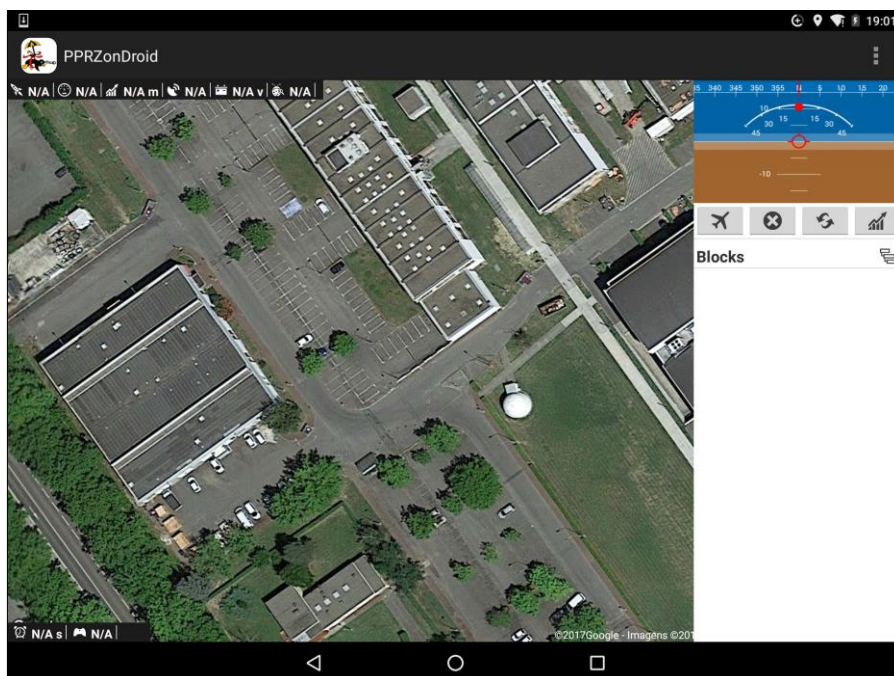


Figura 27 – *Tablet Google HTC Nexus 9* executando o aplicativo *PPRZonDroid*

Fonte: Autor

Na tabela 1 é apresentado um quadro-resumo com as principais características e especificações do *tablet Google HTC Nexus 9*.

<b>Sistema Operacional</b>	Android 5.0
<b>Tela</b>	IPS LCD de 8,9 polegadas com proteção Gorilla Glass 3
<b>Resolução</b>	2048 x 1536 pixels
<b>Densidade de pixels</b>	281 ppi
<b>Chipset</b>	NVIDIA Tegra K1
<b>CPU</b>	Dual core de 2,3 GHz
<b>GPU</b>	Kepler DX1
<b>RAM</b>	2 GB
<b>Armazenamento</b>	16 GB (Wi-Fi) ou 32 GB (Wi-Fi e 4G) – não expansíveis
<b>Câmera traseira</b>	8 megapixels (com foco automático e flash LED)
<b>Câmera frontal</b>	1,6 megapixels
<b>Bateria</b>	6700 mAh (não removível)
<b>Dimensões</b>	22,8 cm (altura) x 15,3 cm (largura)
<b>Espessura</b>	0,8 mm
<b>Peso</b>	425 g
<b>Preço nos EUA</b>	US\$ 349,99 (16 GB) e US\$ 429,99 (32 GB)

Tabela 1 – Principais especificações do *Tablet Google HTC Nexus 9*

Fonte: Disponível em: <<https://img1.ibxk.com.br/2015/01/16/16130341940473.jpg?w=700>>. Data do acesso (Mai, 15), 2017

*Tablets* com configuração igual ou superior à indicada na tabela 1, além de versões mais novas da placa *Raspberry Pi* deverão ser capazes de rodar o sistema projetado neste trabalho sem maiores problemas.

O sistema projetado pode ser utilizado em diversos tipos de VANT, como veículos multirrotores e de asa-fixa, porém, para a realização dos experimentos deste trabalho foi construído um veículo do tipo multirotor. Na figura 28 é apresentado um protótipo do VANT (quadroter) utilizado nos testes deste projeto. Este VANT foi montado pela Equipe *Black Bee Drones* [49] (equipe de VANTs da UNIFEI). Detalhes sobre os principais componentes do VANT e a sua montagem foram apresentados no Capítulo 3.



Figura 28 – VANT (quadrotor) utilizado no projeto

Fonte: Autor

Na figura 29 é possível observar alguns dos principais componentes da estação de controle desenvolvida. É importante ressaltar que os itens de “d” a “g” são utilizados apenas para facilitar a visualização dos dados processados na *Raspberry Pi*, porém não são essenciais para o sistema projetado. Além disso, conforme mencionado no capítulo 3, é necessário que um dos *modems* e a sua placa FTDI (figura 24) sejam conectados a uma das portas USB da placa *Raspberry Pi*. O *hardware* presente na figura 29 é composto por:

- a) *Tablet Google HTC Nexus 9, 2x NVIDIA Tegra K1 @ 2,50 GHz – Sistema Operacional Android Lollipop (versão 5.0.2);*
- b) *Placa de desenvolvimento Raspberry Pi Modelo B+ V1.2 (2014);*
- c) *Adaptador dongle Wi-Fi;*
- d) *Teclado USB;*
- e) *Mouse USB;*
- f) *Conversor HDMI para VGA;*
- g) *Monitor VGA.*



Figura 29 – Esquema mostrando alguns dos componentes da GCS desenvolvida

Fonte: Autor

O *link* de comunicação é vital para o funcionamento correto da estação de controle, pois é através dele que são transmitidos e recebidos todos os dados da aeronave. Como a aplicação deste projeto é para a operação de pequenos VANTs, alguns dos principais objetivos do projeto são:

- *Link* de comunicação estável;
- Baixo consumo energético, pois existe a limitação das baterias no VANT;
- Pequenas dimensões, dadas as limitações físicas e a portabilidade da estação de controle, e também o tamanho dos VANTs.

O projeto desenvolvido visa a portabilidade, facilitando trabalhos de campo com VANTs. Pensando na portabilidade, a ideia básica é fazer a substituição do computador (*notebook*) que seria utilizado em uma GCS tradicional por uma placa *Raspberry Pi* (1). O *Paparazzi UAV* instalado nesta placa fica encarregado de promover o planejamento e o controle de voo do VANT. Após a configuração inicial e estabelecimento da comunicação do *Paparazzi UAV* com o *PPRZonDroid* (aplicativo



instalado no *tablet*) (2), todas as demais configurações e mesmo o controle do voo do VANT em tempo real podem ser feitos através do *tablet* (3). Esse processo pode ser verificado no fluxograma apresentado na figura 30.

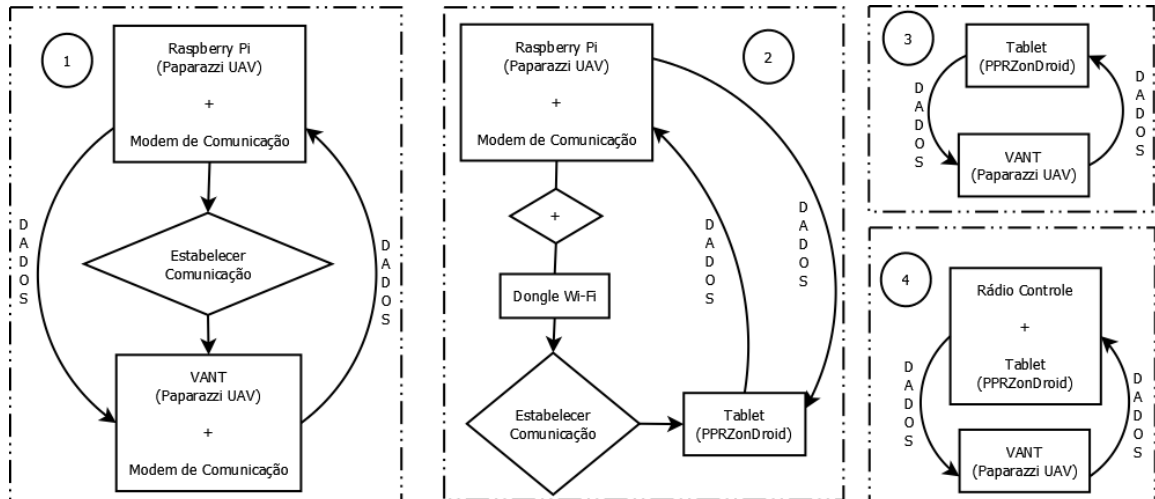


Figura 30 – Fluxograma com a ideia básica deste projeto

Fonte: Autor

A opção (4) demonstrada na figura 30 corresponde a configuração na qual o operador humano toma o controle do VANT através do uso de um rádio controle (RC). Uma boa opção de montagem dos equipamentos para tornar possível esta configuração é demonstrada na figura 31, na qual um suporte é anexado ao rádio controle e o *tablet* é colocado neste suporte, permitindo assim uma maior facilidade no manuseio do rádio controle, além da visualização no *tablet* das informações do voo em tempo real, além de outras aplicações com os dados capturados através do VANT como, por exemplo, a filmagem através de uma câmera acoplada ao VANT. É possível também visualizar no *tablet* eventuais mudanças que o operador faça no plano de voo a partir do RC.



Figura 31 – Suporte para facilitar o manuseio de um *tablet* e um RC

Fonte: Disponível em: <[https://http2.mlstatic.com/suporte-fpv-de-tablet-e-telefone-para-radio-D\\_NQ\\_NP\\_581621-MLB20803951151\\_072016-O.webp](https://http2.mlstatic.com/suporte-fpv-de-tablet-e-telefone-para-radio-D_NQ_NP_581621-MLB20803951151_072016-O.webp)>. Data do acesso (Mai, 13), 2017

## 5 RESULTADOS E DISCUSSÃO

Desde a sua concepção, em virtude de seu tema, este projeto apresenta relevância, uma vez que a utilização e pesquisas sobre VANTs têm se difundido largamente em todo o mundo. Além disso, o investimento financeiro necessário para implementar o sistema proposto é relativamente baixo, pois ele é composto por poucos componentes e todos tendo baixo custo.

A implementação deste projeto é de baixo custo se comparado ao investimento necessário para implementar uma estação de controle em terra tradicional. A placa *Raspberry Pi* usada no projeto custa aproximadamente US\$ 30,00 [50] e o *tablet* utilizado custa aproximadamente US\$ 350,00 [51], resultando em um total de menos de US\$ 400,00. Este valor é significativamente menor que o que seria requerido para comprar um computador (*notebook*) que possua características computacionais similares às apresentadas pelo *tablet* e pela *Raspberry Pi* que são usados neste trabalho. Além disso, pode ser ressaltada como grande vantagem a duração da bateria do *tablet* utilizado, que, segundo o fabricante, pode chegar a até 9,5 horas para utilização com rede *Wi-Fi* e até 30 dias em *standby* [52], além das suas boas especificações computacionais (processamento, memória, processamento gráfico, etc.).

Apesar de haver várias pesquisas em andamento sobre UAVs, este projeto aborda a criação de um sistema inovador, pois a combinação dos componentes de *software* e *hardware* nele utilizados ainda possui poucos trabalhos relacionados e, ainda menos, quando do início de desenvolvimento deste projeto. Especificamente, no Brasil, apesar de haver cada vez mais pesquisas sobre VANTs, este objeto de estudo ainda é relativamente novo, e, portanto, trabalhos similares são realizados em poucos institutos e universidades do País, destacando-se, neste sentido, o ITA (Instituto Tecnológico da Aeronáutica). Deste modo, todas as soluções para os problemas que foram enfrentados ao longo da elaboração deste projeto demandaram esforços consideráveis, por não haver uma vasta gama de referências técnicas para consulta.

Exemplificando as dificuldades encontradas, a determinação do sistema operacional que seria utilizado na *Raspberry Pi* foi feita após várias tentativas com várias versões dos sistemas operacionais *Raspbian* e *Linux*. Conforme pode ser

verificado em [53], o processo de instalação e configuração do *Paparazzi UAV* numa *Raspberry Pi* demora várias horas, podendo levar de 4 a 5 horas. Deste modo, cada vez que alguma tentativa de utilização do *Paparazzi* em algum dos sistemas operacionais testados falhava, seja por erros ao longo da instalação ou mesmo durante a utilização no projeto, uma nova tentativa com outro SO demandava quantidade de tempo semelhante. Todas essas tentativas serviram para auxiliar a comunidade de usuários do *Paparazzi* uma vez que as falhas foram comunicadas aos desenvolvedores do Sistema, os quais adicionaram informações sobre esses erros em algumas páginas do *site* do *Paparazzi*, alertando sobre tais erros, o que é demonstrado em [53]. A informação “*This page only describes the installation on Raspberry Pi 1. If you are running Ubuntu on the Raspberry Pi 2, follow the normal Installation/Linux instructions*” (Esta página descreve somente a instalação na *Raspberry Pi 1*. Se você está executando o *Ubuntu* na *Raspberry Pi 2*, siga as instruções normais de instalação no *Linux*) foi acrescentada após os erros na instalação do *Paparazzi UAV* na placa *Raspberry Pi* usada neste trabalho terem sido comunicados aos desenvolvedores do *Paparazzi* (anteriormente esta informação não constava na página de instalação).

Além das dificuldades anteriormente citadas, como o *Paparazzi UAV* é um sistema de código aberto, a maior parte do auxílio encontrado para solucionar erros em sua utilização vem da sua comunidade de usuários e desenvolvedores. Em alguns casos, achar a solução para problemas encontrados nem sempre foi rápida ou mesmo trivial. Um exemplo desta dificuldade em solucionar problemas neste tipo de sistema foi encontrado ao tentar realizar a comunicação entre o aplicativo *PPRZonDroid* e o *software Paparazzi UAV*. Na grande maioria dos casos ocorria falha de *socket* e o servidor da aplicação parava, encerrando a comunicação, sem que houvesse um motivo facilmente identificável para tal ocorrência. Para tentar solucionar esse erro foi preciso avaliar profundamente o código-fonte do projeto trabalhando em parceria com os desenvolvedores do sistema *Paparazzi*, uma vez mais gerando informações em fóruns que podem ser utilizados pela comunidade de usuários do *Paparazzi*.

Por fim, ressaltando-se a relevância do projeto desenvolvido, um artigo que foi escrito tendo como base este trabalho foi submetido e aceito para publicação no Congresso ITNG 2017 (*14th International Conference on Information Technology: New Generations*), realizado em abril de 2017 em Las Vegas, Nevada, EUA.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Permitir que a estação de controle de um MiniVANT seja implantada em um *tablet* garantirá ao usuário uma enorme gama de funcionalidades que são oferecidas por este tipo de equipamento, como a mobilidade, a ergonomia e a facilidade para serviços de campo, além da possibilidade da sua utilização em vários locais e situações e para diversas finalidades. Além disso, o sistema projetado possui relevância tanto para aplicações civis quanto militares.

Após o término deste trabalho pode-se concluir que é possível montar este tipo de modelo totalmente autônomo, e mais importante ainda, com um baixo custo utilizando-se tecnologias de código aberto e componentes mais baratos do que os que seriam utilizados tradicionalmente em projetos similares.

Este trabalho preza pelo desenvolvimento de um sistema portátil e de baixo custo que permite ao usuário realizar, em diversos ambientes, várias atividades com o seu VANT. Após fazer a configuração inicial, a inicialização do servidor de aplicação e o estabelecimento da comunicação entre o *Paparazzi UAV* e o *PPRZonDroid*, todas as demais configurações e o controle de voo poderão ser realizadas em tempo real através do *tablet*.

Este projeto já está sendo utilizado como ponto de partida para outros projetos similares e, além disso, pode ser usado como base para diversos trabalhos futuros, por exemplo: a implementação de aplicativos para serem utilizados no *tablet* promovendo um melhor controle do VANT ou mesmo usando os dados capturados no voo para as mais diversas finalidades; a substituição de um ou mais componentes deste sistema tentando melhorar a sua eficiência; implementar um sistema ainda mais portátil no qual a GCS seja totalmente implantada num *tablet*, etc.

## Referências

- [1] "Força Aérea esclarece normas para voos de VANTs no Brasil". 2016. [on line]. Disponível em: <<http://www.brasil.gov.br/defesa-e-seguranca/2015/03/forca-aerea-esclarece-normas-para-voos-de-VANTs-no-brasil>>. Data do acesso (Nov, 18), 2016.
- [2] Sharkey, N. The Automation and Proliferation of Military Drones and the Protection of Civilians. Publicado em: Law, Innovation and Technology, 2011.
- [3] Golightly, I. e Jones, D. Visual control of an unmanned aerial vehicle for power line inspection. Publicado em: Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on, páginas 288–295. IEEE.
- [4] Thompson II, R. M. Drones in Domestic Surveillance Operations: Fourth Amendment Implications and Legislative Responses. Publicado em: Congressional Research Service, 2013.
- [5] Tripicchio, P., Satler, M., Dabisias, G., Ruffaldi, E., Avizzano, C. A. Towards Smart Farming and Sustainable Agriculture with Drones. Publicado em: Intelligent Environments (IE), 2015 International Conference on, 2015.
- [6] Finn, R. L., Wright, D. Unmanned aircraft systems: Surveillance, ethics and privacy in civil applications, Computer Law & Security Review, Volume 28, Issue 2, 2012, Pages 184-194, ISSN 0267-3649.
- [7] Paneque-Gálvez, J., McCall, M. K., Napoletano, B. M., Wich, S. A., Koh, L. P. Small Drones for Community-Based Forest Monitoring: An Assessment of Their Feasibility and Potential in Tropical Areas. Forests, 2014, 5(6), 1481-1507; doi:10.3390/f5061481.
- [8] Gregory, D. From a View to a Kill: Drones and Late Modern War. Volume: 28 issue: 7-8, page(s): 188-215. Article first published online: January 12, 2012; Issue published: December 1, 2011. DOI: <https://doi.org/10.1177/0263276411423027>.
- [9] Jorge, L. A. C., Inamasu, R. Y. Uso de veículos aéreos não tripulados (VANT) em Agricultura de Precisão. Embrapa Instrumentação, Ferramentas para Agricultura de Precisão, 2014.
- [10] Santos, C. O futuro do agronegócio brasileiro. Revista Conjuntura Econômica, 2014.
- [11] Cosentino, A., Smiderle, C. D., Homma, R. Z., Fraco, R. A., Bose, R., Sohn, O. A., Szymanski, C. Desenvolvimento de Metodologia e Sistema para Tratamento de Imagem e Reconhecimento de Padrões para Inspeção Autônoma de Linhas Aéreas de Distribuição e Transmissão. XXII Seminário Nacional de Distribuição de Energia Elétrica, 2016.

- [12] Faria, R. R., Costa, M. E. A inserção dos veículos aéreos não tripuláveis (drones) como tecnologia de monitoramento no combate ao dano ambiental. *Revista Ordem Pública* v. 8, n. 1, 2015.
- [13] Albiero, V., De Biasi, H. H. Construção de Mosaico Utilizando Imagens Aéreas Adquiridas de Forma Autônoma. *Unoesc & Ciência ACET* v. 7, n. 1 (2016), Editora Unoesc. ISSN: 2178-342X.
- [14] Angonese, A. T., Rosa, P. F. F., & Tibúrcio, P. G. Estação de Controle em Solo com Funcionalidades de Voo Múltiplo para VANTs. *Simpósio Brasileiro de Automação Inteligente*, 2014.
- [15] Kang, W.; Yuan, M. Software Design for Mini-type Ground Control Station of UAV. School of Automation Science and Electrical Engineering, Beijing University of Aeronautics & Astronautics, 2009.
- [16] Amorim, L. A., Santos, M. C., de Queiroz, F. M., Silva, L. D. A., Vassalo, R. F., & Sarcinelli-Filho, M. Estimação de Posição e Altitude de um Veículo Aéreo não Tripulado baseada em GPS, IMU e Dados Visuais.
- [17] Austin, R. *Unmanned Aircraft Systems: UAVS design, development and deployment*. [S.l.]: John Wiley & Sons, 2010.
- [18] "ArduPilot :: About". 2017. [on line]. Disponível em: <<http://ardupilot.org/about>>. Data do acesso (Fev, 18), 2017.
- [19] "Arduino - Home". 2017. [on line]. Disponível em: <<https://www.arduino.cc/>>. Data do acesso (Fev, 18), 2017.
- [20] Patelli, A., & Mottola, L. (2016, June). Model-based real-time testing of drone autopilots. In *Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use* (pp. 11-16). ACM.
- [21] Bin, H., & Justice, A. (2009). The design of an unmanned aerial vehicle based on the ArduPilot. *Indian Journal of science and Technology*, 2(4), 12-15.
- [22] "PaparazziUAV". 2016. [on line]. Disponível em: <[http://wiki.paparazziuav.org/wiki/Main\\_Page](http://wiki.paparazziuav.org/wiki/Main_Page)>. Data do acesso (Nov, 16), 2016.
- [23] "PPRZonDroid". 2016. [on line]. <<http://wiki.paparazziuav.org/wiki/PPRZonDroid>>. Data do acesso (Nov, 16), 2016.
- [24] CROTTI, Y. et al. RaspberryPi e Experimentação Remota. *ICBL2013. International Conference on Interactive Computer aided Blended Learning*, 2013.
- [25] Garrett, F. "Como funciona o Raspberry Pi? Entenda a tecnologia e sua aplicabilidade". 2016. [on line]. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2014/11/como-funciona-o-raspberry-pi-entenda-tecnologia-e-sua-aplicabilidade.html>>. Data do acesso (Nov, 16), 2016.

- [26] Upton, E., & Halfacree, G. (2014). Raspberry Pi user guide. John Wiley & Sons.
- [27] Vijayakumar, N., & Ramya, R. (2015, March). The real time monitoring of water quality in IoT environment. In Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference on (pp. 1-4). IEEE.
- [28] Benemann, A. (2013). Estação de controle para veículos aéreos não tripulados.
- [29] "COMUNICAÇÃO XBEE - IFSC São José". 2017. [on line]. Disponível em: <[http://wiki.sj.ifsc.edu.br/wiki/index.php/COMUNICA%C3%87%C3%83O\\_XBEE](http://wiki.sj.ifsc.edu.br/wiki/index.php/COMUNICA%C3%87%C3%83O_XBEE)>. Data do acesso (Abr, 27), 2017.
- [30] Kinney, P. (2003, October). Zigbee technology: Wireless control that simply works. In Communications design conference (Vol. 2, pp. 1-7).
- [31] Lecheta, R. R. (2013). Google Android-3ª Edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. Novatec Editora.
- [32] Rogers, R., Lombardo, J., Mednieks, Z., & Meike, B. (2009). Android application development: Programming with the Google SDK. O'Reilly Media, Inc.
- [33] "Paparazzi Center - Paparazzi UAV". 2017. [on line]. Disponível em: <[https://wiki.paparazziuav.org/wiki/Paparazzi\\_Center](https://wiki.paparazziuav.org/wiki/Paparazzi_Center)>. Data do acesso (Fev, 18), 2017.
- [34] "GCS - Paparazzi UAV". 2017. [on line]. Disponível em: <<https://wiki.paparazziuav.org/wiki/GCS>>. Data do acesso (Fev, 18), 2017.
- [35] "Category:Autopilots - Paparazzi UAV". 2017. [on line]. Disponível em: <<https://wiki.paparazziuav.org/wiki/Category:Autopilots>>. Data do acesso (Fev, 18), 2017.
- [36] "Lisa/M v2.0 - Paparazzi UAV". 2017. [on line]. Disponível em: <[https://wiki.paparazziuav.org/wiki/Lisa/M\\_v2.0](https://wiki.paparazziuav.org/wiki/Lisa/M_v2.0)>. Data do acesso (Fev, 18), 2017.
- [37] "Lisa/M v2.0 - Paparazzi UAV - Pinout". 2017. [on line]. Disponível em: <[https://wiki.paparazziuav.org/wiki/Lisa/M\\_v2.0#Pinout](https://wiki.paparazziuav.org/wiki/Lisa/M_v2.0#Pinout)>. Data do acesso (Fev, 18), 2017.
- [38] Daponte, P., De Vito, L., Mazzilli, G., Picariello, F., Rapuano, S., & Riccio, M. (2015, June). Metrology for drone and drone for metrology: Measurement systems on small civilian drones. In Metrology for Aerospace (MetroAeroSpace), 2015 IEEE (pp. 306-311). IEEE.
- [39] "AspirinIMU - Paparazzi UAV". 2017. [on line]. Disponível em: <<https://wiki.paparazziuav.org/wiki/AspirinIMU>>. Data do acesso (Fev, 18), 2017.
- [40] Floreano, D., & Wood, R. J. (2015). Science, technology and the future of small autonomous drones. Nature, 521(7553), 460.

- [41] Gati, Balazs. "Open source autopilot for academic research-the paparazzi system." American Control Conference (ACC), 2013. IEEE, 2013.
- [42] Felizardo, L. F. (2016). Integração de Software e Hardware no Desenvolvimento de VANTs Autônomos de Baixo Custo.
- [43] "Github". 2017. [on line]. Disponível em: <<https://github.com/>>. Data do acesso (Mai, 11), 2017.
- [44] "Git". 2017. [on line]. Disponível em: <<https://git-scm.com/>>. Data do acesso (Mai, 11), 2017.
- [45] Glover, J. M. "VANT University". Editora: VANTUniversity (2014). ISBN-10: 0692316035; ISBN-13: 978-0692316030.
- [46] Barnhart, R. K., Hottman, S. B., Marshall, D. M. "Introduction to unmanned aerial systems". Editora: CRC Press (2011). ISBN-10: 1439835209; ISBN-13: 978-1439835203.
- [47] Fahlstrom, P. G., Gleason, T. J. "Introduction to uav systems". Editora: Uav Systems Inc (1993). ISBN-10: 9995144328; ISBN-13: 978-9995144326.
- [48] "Installation - PaparazziUAV". 2016. [on line]. Disponível em: <<https://wiki.paparazziuav.org/wiki/Installation>>. Data do acesso (Out, 10), 2016.
- [49] "Black Bee Drones". 2017. [on line]. Disponível em: <<https://blackbee.unifei.edu.br/>>. Data do acesso (Fev, 18), 2017.
- [50] "Raspberry Pi Model B+". 2016. [on line]. Disponível em: <<http://uk.rs-online.com/web/p/processor-microcontroller-development-kits/8111284/>>. Data do acesso (Dez, 11), 2016.
- [51] "BUY THE NEXUS 9 – HTC United States". 2016. [on line]. Disponível em: <<http://www.htc.com/us/go/buy-now-nexus-9/>>. Data do acesso (Dez, 11), 2016.
- [52] "Google Nexus 9 Specs and Reviews". 2016. [on line]. Disponível em: <<http://www.htc.com/us/tablets/nexus-9/>>. Data do acesso (Dez, 11), 2016.
- [53] "Installation/RaspberryPi - PaparazziUAV". 2016. [on line]. Disponível em: <<https://wiki.paparazziuav.org/wiki/Installation/RaspberryPi>>. Data do acesso (Dez, 11), 2016.