

DESENVOLVIMENTO DE SISTEMA DE CONTROLE PARA VEÍCULO AÉREO NÃO TRIPULADO (VANT) POR MEIO DE DISPOSITIVOS MÓVEIS

Célia da Silva Morais¹,

Alex Coelho²,

Igor Yepes³

RESUMO

Conceitos de mobilidade aplicados aos dispositivos móveis, seguidos pela evolução em campos importantes da tecnologia como visão computacional e inteligência artificial, compõem um vasto leque a ser explorado, como no caso dos VANTs (Veículos Aéreos Não Tripulados). Nesse contexto, este trabalho propõe a implementação de um software para dispositivo móvel que possibilite o controle de um VANT por meio de comunicação sem fio. Foram realizados estudos de trabalhos relacionados para compreensão da aerodinâmica dessas aeronaves. O ponto principal do trabalho se ateve a comunicação do dispositivo móvel com a plataforma de hardware Arduino por meio de protótipos divididos entre simulações variadas de comunicação USB e sem fio. O software proposto é composto por dois joysticks que permitem o envio de até dezesseis comandos para a plataforma Arduino, incluindo ainda demais comandos importantes ao controle da aeronave e acompanhamento de informações importantes na tela.

Palavras-chave: : Dispositivos Móveis, VANT, Sistema de Controle.

ABSTRACT

Mobility concepts apply to mobile devices followed by developments in important fields of technology as computer vision and artificial intelligence, composing a vast range to be explored is the case with the UAV (Unmanned Aerial Vehicles). In this context, the work proposes the implementing one software for a mobile device that enables the control of a UAV by a wireless. Research of related work was performed to understand the aerodynamics of these aircraft. The principal point for these tests was the communication of the mobile device with Arduino, through prototypes divided between

¹Acadêmica do Curso de Sistemas de Informação - Fundação Universidade do Tocantins (UNITINS) - e-mail: cdamorais@hotmail.com

²Professor do Curso de Sistemas de Informação - Fundação Universidade do Tocantins (UNITINS) - e-mail: alex.c@unitins.br

³Professor do Curso de Sistemas de Informação - Fundação Universidade do Tocantins (UNITINS) e-mail: igor.y@unitins.br

simulations of USB and wireless communication. The software is composed by two joysticks in the principal screen that allows sending sixteen commands to control the Arduino platform and monitoring of important information on the screen.

Key-Words: Mobile Device, UAV, Control System.

INTRODUÇÃO

A evolução da tecnologia móvel é responsável pelo avanço em diferentes aspectos da mobilidade. Essa evolução é possível, dentre outros fatores, graças ao avanço no universo de aplicações móveis direcionadas às interações em tempo real. Diante desse fato, a ideia de "Estação de Trabalho", ou mesmo de "Desktop" passa a ser irrelevante para o mundo atual, visto a possibilidade da realização de muitas atividades por meio de dispositivos móveis. A mobilidade hoje é uma necessidade primordial no cotidiano, em que se espera acesso imediato às novas soluções apresentadas com o fim de possibilitar tomadas de decisões com a máxima flexibilidade e em tempo hábil (BRASSCOM, 2013).

Neste contexto de possibilidades diversas de aplicações para dispositivos móveis, sua utilização como dispositivo de controle já é observada no contexto residencial, também conhecido como Domótica, e mesmo no âmbito comercial, com destaque para soluções de mobilidade. Essa concepção especial dos dispositivos móveis aliada a outros campos tecnológicos permite um estudo abrangente na área, subsidiando outras estruturas como a dos VANTs (Veículos Aéreos Não Tripulados).

Pesquisas na área de modelagem dinâmica e desenvolvimento de métodos de controle automático para estabilização e regulação de voo para VANTS ganham espaço e destaque atualmente devido à complexidade existente nesse âmbito (SÁ, 2012). Como exemplo dessa complexidade está o desenvolvimento do sistema de controle para o VANT, que se baseia na problemática do mapeamento entre as variáveis que definem o comportamento da aeronave, com sua interface de manipulação realizada por software.

Frente a essas considerações verifica-se que a integração de plataformas e instrumentos tecnológicos podem concretizar possibilidades diferenciadas para que sejam enfrentadas e trabalhadas nas mais diversas áreas do conhecimento, com a utilização de sistemas embarcados em VANTs controlados por tablets ou smartphones em redes sem fio.

O presente trabalho de pesquisa propõe o desenvolvimento de um software para dispositivo móvel que possibilite o controle remoto para VANTs com base na plataforma

Arduino, por meio de ambientes de redes sem fio, e que permita voos controlados por um operador humano. Os objetivos específicos envolvem a compreensão dos conceitos aerodinâmicos associados aos VANTS do tipo quadrotor, levantamento de requisitos, desenvolvimento e testes do sistema.

1. METODOLOGIA

· Para atender as condições do problema exposto, foram realizadas as seguintes atividades:

- Levantamento de requisitos
- Estudo sobre o VANT e suas diversas aplicabilidades.
- Estudos sobre a aerodinâmica do VANT do tipo quadrotor.
- Comunicação entre o sistema operacional android e a plataforma Arduino, base de construção do VANT.

- Estudo das bibliotecas e protocolos de redes indispensáveis para o desenvolvimento do software.

- Criação de protótipos com a plataforma Arduino.
- Criação do modelo de interface gráfica do controle e implementação do software.

- Realização de testes por meio de simulação nos protótipos criados.
- As atividades citadas foram realizadas simultaneamente, no decorrer do trabalho, pois a cada passo dos estudos eram realizados testes com base na comunicação via cabo USB e comunicação através de radio frequência. Para tanto, foi utilizado um tablet com conexão host-USB, placa APC220 para comunicação de radiofrequência, o microcontrolador Arduino e acessórios para montagem de protótipos.

Para desenvolver o aplicativo em Android adotou-se a linguagem Java com suas ferramentas indispensáveis: A IDE(Integrated Development Environment) do Eclipse e o SDK (Kit de Desenvolvimento de Software) recomendado para desenvolvedores Android. Já para simulações no Arduino, a linguagem utilizada foi c e c++.

2. VANT - VEÍCULO AÉREO NÃO TRIPULADO

O termo VANT se refere a todo tipo de aeronave que não precise de um piloto embarcado. São aeronaves ideais para realização de atividades exaustivas para um ser humano ou em locais de difícil acesso.

Segundo Vasconcelos (2013), o VANT tem capacidade de possuir autocontrole

através da implementação do sistema com dispositivos de rádio frequência. Porém, para uma abordagem diferente, é interessante a utilização do controle remoto, tanto para controlar melhor o veículo, como para aumentar o desempenho do mesmo. Observa-se que de modo geral, o sistema de piloto automático possui um computador central que determina seus movimentos. De acordo com Oliveira (2005), o sensor de posição ou movimento define a posição atual da aeronave e conforme os dados informados pelo piloto ou por um computador de navegação realiza-se a correção necessária através dos servo motores e atuadores.

Dentre os modelos de asas rotativas, o quadrotor se destaca cada vez mais em pesquisas e desenvolvimento. A Figura 1 mostra um modelo do VANT do tipo quadrotor.



Figura - VANT do tipo quadrotor

FONTE:(BENEMANN, 2013)

Segundo Sá (2012), um quadrotor é uma aeronave dotada de quatro asas rotativas e seus motores são posicionados de forma que o empuxo gire na mesma posição, o que permite um voo vertical, estagnado e de baixa velocidade com poder de manobras diversas.

2.1. ANDROID

O Android é um Sistema Operacional completo baseado em Linux com capacidade diversificada para sua utilização. O uso de aparelhos com o sistema Android tornou-se popular, e a disseminação de aplicativos favorece às diversas aplicabilidades que os aparelhos possuem.

A tecnologia Android oferece uma plataforma para desenvolvimento inovadora e flexível para dispositivos móveis. Uma das vantagens do uso do Android está na facilidade que existe desde a aquisição dos aparelhos até a publicação de aplicativos no Google play. A escolha da linguagem Android favorece a implementação devido ser

uma linguagem com base em Java, ao qual faz parte das disciplinas oferecidas pelo curso, pode ser desenvolvida em uma IDE(Integrated Development Environment) de fácil manipulação e é compatível com quase todos os aparelhos disponíveis no transcorrer do projeto (ANDROID, 2014).

A partir do nível API12 do Sistema Operacional Android está disponibilizado a biblioteca que trata todos os pontos importantes para conexão USB, o que auxilia na criação e utilização de drivers adicionais para implementação da comunicação entre o dispositivo móvel e o Arduino (ANDROID, 2014). O Quadro 1 traz as principais classes de comunicação nativas do Android que foram aplicadas a esse projeto.

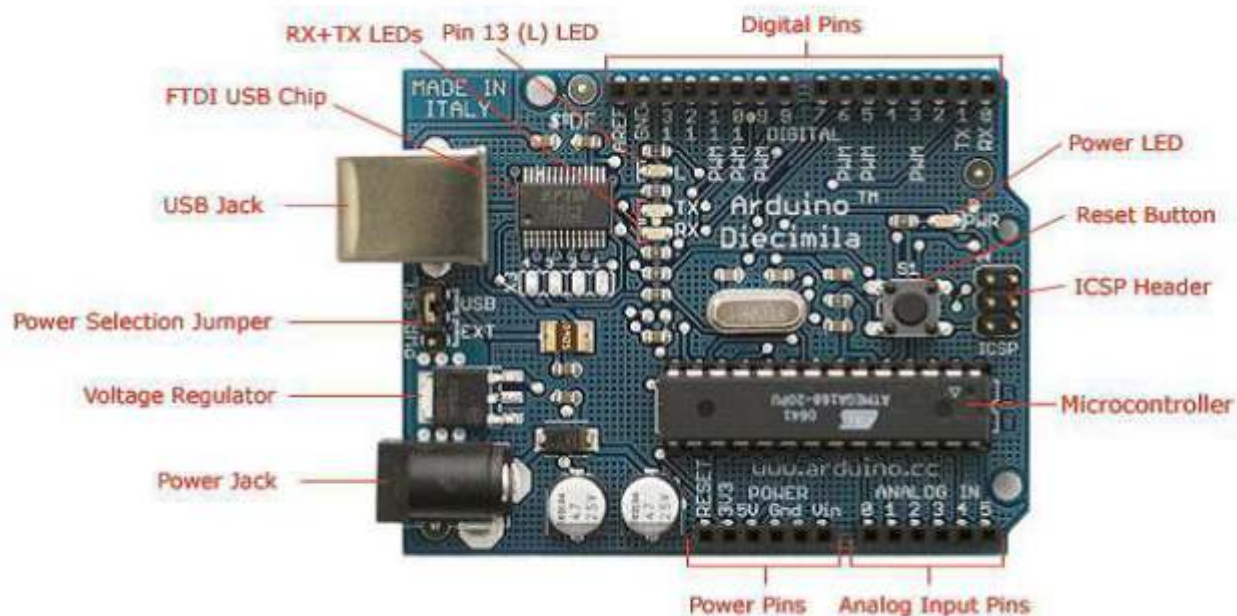
Quadro 1 - Principais classes de comunicação USB Android

QUADRO I	
CLASSE	DESCRIÇÃO
UsbManager	Acessa o estado do USB e se comunica com os periféricos conectados.
UsbDevice	Faz a comunicação com o hardware periférico, desde que o dispositivo Android esteja com host USB.
UsbAccessory	Representa um acessório USB.
UsbDeviceConnection	Envia e recebe as mensagens de dados e de controle para o dispositivo USB.
UsbRequest	Representa uma solicitação

Esses recursos não são suficientes para estabelecer a conexão com a plataforma Arduino e interpretar seus dados em uma comunicação. Para isso foi assumida a biblioteca UsbSerialLibrary, que traz todas as soluções relacionadas à comunicação por meio da porta Serial, seja via USB direto, ou radiofrequência. O uso da biblioteca UsbSerialLibrary é livre sob os termos da GNU Lesser General Public onde é explícito o direito de utilização, modificação e redistribuição (GNU, 2014).

2.2. ARDUINO

O Arduino é uma plataforma de fundamental importância para criação de protótipos devido o software ser desenvolvido para controlar um VANT com base em Arduino. Trata-se de uma placa de microcontrolador, originado na Itália em 2005. O objetivo principal é criar dispositivos para controlar projetos e protótipos com o menor custo possível que pode ser utilizada para projetos ligados ao computador e mesmo para objetos interativos. Um modelo de Placa Arduino com descrição das principais partes é apresentado na Figura 2.



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Figura 2 - Plataforma Arduino FONTE: (BRESSAN, 2014)

O Arduino possui 6 entradas analógicas em que cada uma se liga a um conversor analógico-digital de 10 bits, o que permite que a leitura analógica se transforme em um valor entre 0 e 1024 . Sua voltagem vai de 0 a 5V, o que pode ser alterado conforme a necessidade da aplicação (BRESSAN, 2014). As principais características do Arduino estão apresentadas no Quadro 2.

Quadro 2 - Características do Arduino

QUADRO II	
NOME	CARACTERÍSTICA
Microcontrolador	Atmega328 ou ATmega168
Tensão de alimentação	7 - 12 V
Pinos I/O digitais	14(6 podem ser saídas PWM)
Pinos de entrada analógica	6
Corrente contínua do pino I/O	40 mA
Corrente contínua para o pino 3.3 V	50 mA
Memória flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Frequência de clock	16 MHz

Fonte dos dados: (BRESSAN, 2014)

A comunicação entre a Plataforma Arduino e o mundo exterior se dá por meio da porta serial. É possível realizar a comunicação entre o microcontrolador e um programa externo ao ambiente Arduino somente com o cabo USB, o mesmo que usa para ligar o aparelho ao computador, que traduz os bits para a porta serial. Essa comunicação pode

ser feita entre qualquer microcontrolador habilitado para conversas nos parâmetros da comunicação serial (SILVA, 2014).

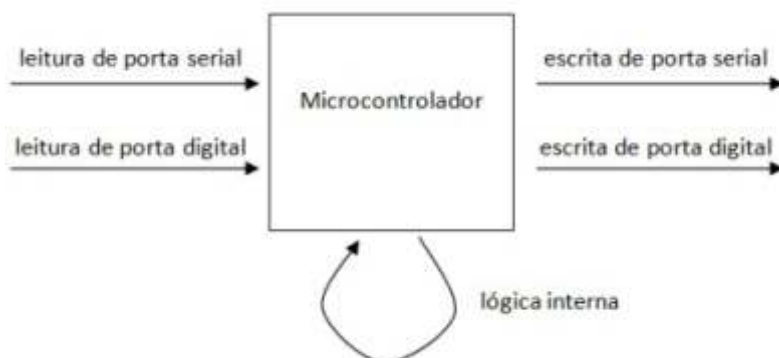


Figura 3 - Exemplo de comunicação pela porta serial - Arduino

Fonte: (STUFF, 2014).

O Arduino possui duas portas separadas para comunicação serial, que são respectivamente as portas 0 e 1, ou seja, a RX(receive) e TX(transfer). É onde acontece toda a transferência de dados (SILVA, 2014).

2.3. COMUNICAÇÃO POR RADIO FREQUÊNCIA

Frequência de rádio, ou ondas de radiofrequência se refere a uma taxa de oscilação no intervalo que vai de cerca de 3KHz a 300 GHz. Essas ondas são campos magnéticos utilizados nas comunicações sem fio. A comunicação se dá com o transporte de energia de um ponto a outro sem a necessidade de fios, como nas transmissões de televisão, rádio e celulares. O sinal que propaga no ar pode ser em forma de ondas ou em linha reta (SENAI, 2014).

2.3.1. APC220

A comunicação entre o dispositivo móvel e a placa Arduino idealizada para o projeto foi feita por meio de radiofrequência que permite a transferência de dados pela porta serial por meio da entrada USB. Para essa comunicação é utilizado o módulo de rádio APC220 que integra a comunicação entre o rádio e a porta serial do Arduino, o que permite uma comunicação transparente para o usuário. Esse módulo possui um microprocessador de alto desempenho capaz de transmitir a uma distancia de até 1000m em linha de visão a 9600 bips, o que o torna perfeito para utilização em VANTs.

A comunicação de radiofrequência só acontece se as duas placas de comunicação estiverem dentro da mesma calibragem de frequência. Para isso,

primeiro é imprescindível que o driver da placa esteja instalado na maquina que fará a calibragem. Alguns computadores realizam a instalação do driver automaticamente, mas se isso não acontecer, o trabalho é feito manualmente e, nesse caso é importante saber qual chip está acoplado no adaptador USB. Para esse projeto, foi utilizado o adaptador USB com chip CP2102, que vem como acessório da placa APC220. O drive para esse chip é disponibilizado pela Silicon Laboratories.



Figura 4 - Transmissor APC220 com acessórios.

Após a instalação do driver, é necessário verificar qual porta USB está ocupada pelo driver, que pode ser averiguado em "gerenciamento de dispositivos" do computador.



Figura 5 - Localização da porta de entrada do driver

Após isso se deve descarregar e instalar o programa que possibilita a calibragem da frequência. Para esse projeto foi utilizado o software RF-Magic 1.2, com a finalidade de calibrar a frequência e ajustes necessários para os testes. Para evitar erros no sistema é necessário que o programa RF-Magic inicie no modo administrador e isso, só depois de o adaptador USB for reconhecido no computador. Ponto interessante,

durante a configurações e que se verificou, consiste no fato de que muitas vezes o computador reconhecia a entrada USB e com isso o software de configuração tinha seu funcionamento comprometido. Isso pode acontecer devido à geração aleatória de porta, e a solução encontrada para esse problema foi fazer alteração da porta diretamente no driver, em gerenciamento de dispositivos, onde se modifica a porta padrão para a menor que esteja disponível no sistema.

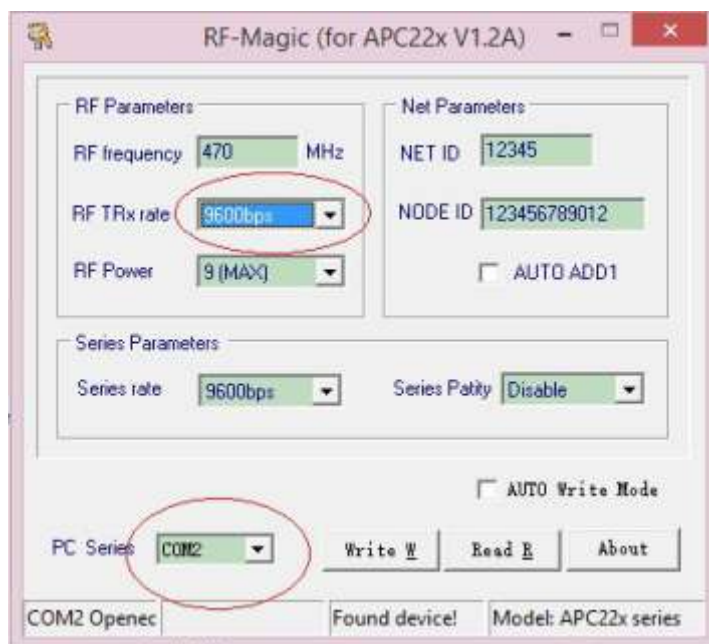


Figura 6 - Janela do software RF-Magic

As principais configurações aqui são a frequência, taxa de séries e taxa de RF-TRx que devem estar iguais nas duas placas e nos programas de testes. Essas configurações foram utilizadas para os testes de comunicação no Arduino tanto para o notebook como para o Tablet. A comunicação diretamente com o VANT acontece por meio de um link de telemetria definidos pelo protocolo MAVLink .

2.3.2. Protocolo de comunicação MAVLINK

Para o projeto foi adotado o protocolo de comunicação MAVLink (Micro Air Vehicle Link) devido esse já estar presente em diversos trabalhos relacionados aos projetos com VANTs como: ARDUPILOT, PX4FMU, SMARTAP, MATRIXPILOT E DROIDPLANNER.

O protocolo MAVLink foi projetado por Lorenz Meier e está liberado para uso sobre a licença LGPL(Free Software Foundation) desde 2009. Sua principal característica é a transferência de dados do VANT, como posição e orientação. Toda

comunicação é realizada por meio de pacotes pré-definidos em um arquivo XML (BENEMANN, 2013).

3. RESULTADOS

O objetivo do sistema de controle foi possibilitar o controle de um VANT do tipo mini quadrotor por meio de um dispositivo móvel com base na plataforma Arduino. Inicialmente, para um melhor entendimento foram levantados os requisitos para o sistema e que são descritos no Quadro 3.

Quadro 3 - Requisitos para o Sistema de Controle Remoto

QUADRO III	
NOME	DESCRIÇÃO
REQUISITOS FUNCIONAIS	Opção de ligar e desligar
	Receber dados e enviar comandos para a aeronave
	Operar manualmente a aeronave através de dois joysticks: Um para altitude e outro para direcionamento
REQUISITOS NÃO FUNCIONAIS	Sistema Operacional – Android versão mínima 3.1.
	Conexão USB-OTG
	Capacidade de conexão através de rede sem fio.
	Modo host-USB .
	IDE(Integrated Development Environment) – Eclipse com Android SDK – API mínima 14.
	RF Magic ou RF-ANET.
	Driver para RF.
	Arduino IDE 1.0.5.
	Linguagens – Java e c++.
	Protocolo de comunicação Serial(RX TX).
Protocolo de comunicação – MAVLink	

Para uma melhor verificação dos resultados foram realizados testes de comunicação entre um computador e o Arduino por meio da porta serial. Para os primeiros testes foi criado um protótipo com 3 leds acionados por um programa simples no Arduino que permite entrada de dados do teclado e um sensor de temperatura que envia essa informação do Arduino por meio da porta serial. A Figura 7 mostra o protótipo do hardware.

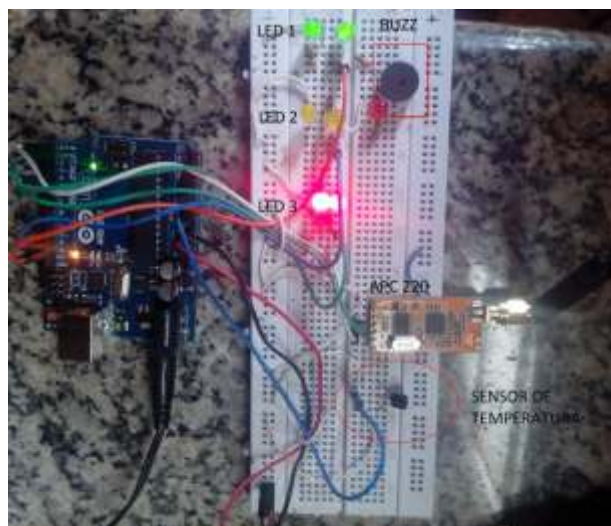


Figura 7 - Protótipo em Arduino para testes de comunicação

Na parte de software para Desktop, a implementação foi feita em Java com a utilização do protocolo RX-TX. O esquema do protótipo final do hardware é apresentado na Figura 8. Nota-se que foi utilizado recursos simples apenas para garantir a sincronização externa entre o Arduino e o computador.

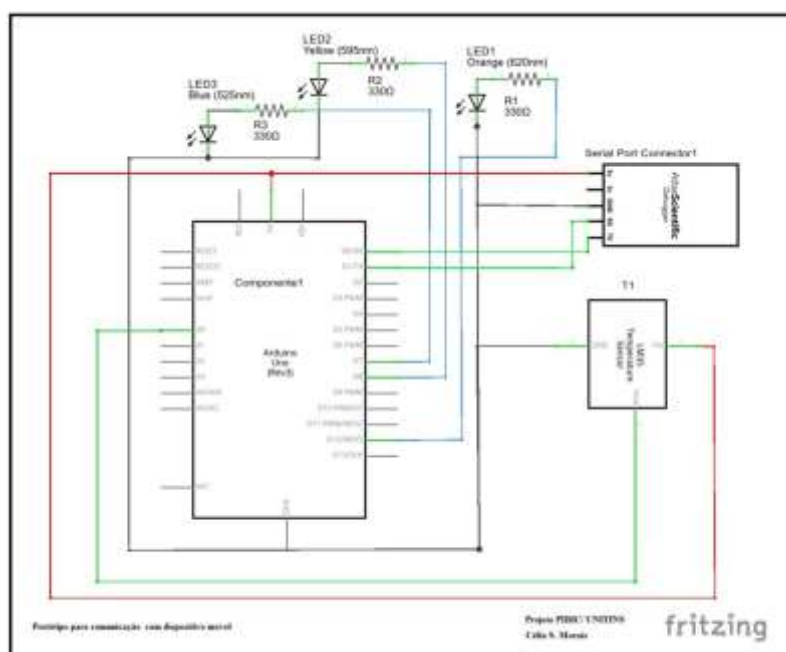


Figura 8 - Protótipo para teste de comunicação via porta Serial

O teste de comunicação tem como objetivo abrir a conexão da porta Serial, reconhecer o dispositivo, enviar os bits de dados e receber uma resposta. Esse primeiro momento de teste sem dispositivo móvel foi importante para definir os parâmetros necessários para a comunicação direta via USB, e posteriormente, via radiofrequência.

De modo geral, o sistema funciona em passos: quando iniciado, a conexão entre as partes é verificada e um feedback no console da máquina diz se houve conexão, e em caso positivo, mostra qual porta USB está ocupada pelo driver. Depois de apresentar os dados recebidos do Arduino pode-se então escolher os botões para ligar ou desligar os leds. Ao clicar em SAIR, o sistema fecha a conexão com a porta Serial. A Figura 9 mostra a tela da aplicação utilizada para testes no computador Desktop.

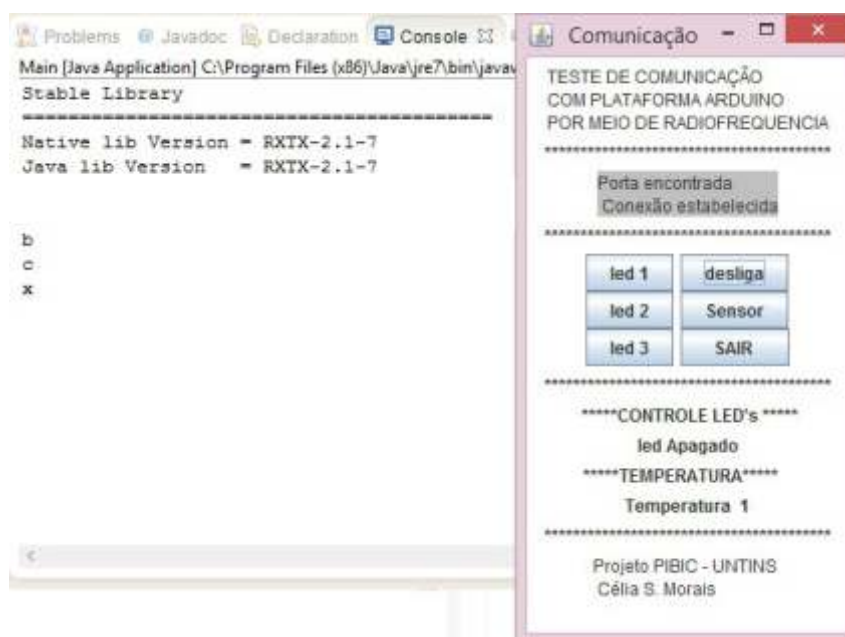


Figura 9 - Teste de comunicação com Arduino

Esse mesmo software para desktop foi utilizado para fazer a comunicação por radiofrequência. A comunicação via porta Serial acontece de forma totalmente transparente para o usuário, tanto que para os teste com o módulo de rádio não houve necessidade de fazer qualquer tipo de alteração no código utilizado. As mudanças aconteceram nas configurações de hardware, em que alguns valores precisam ser igualados em todos os meios que se comunicam, que nesse caso são as duas placas de comunicação, o sistema do Arduino e o software de teste. O teste em ambiente desktop funcionou como preparação para o trabalho direto no dispositivo móvel, que demanda mais estudo devido sua complexidade. Fato é que o computador possui uma interface serial nativa, o que facilita a comunicação USB. Já para o sistema Android, são necessários drivers adicionais e interpretação da API com host USB, que vai funcionar como controle de transferência em massa, porem o sistema Android envia uma string para o conversor, que por sua vez a converte em bytes que o Arduino está esperando. Essa ponte pode causar delays maiores que o alcançado na comunicação direta do

desktop, e precisa ser tratada em código.

Para desenvolvimento do controle, um dos requisitos apresentados é a possibilidade de navegação do VANT, o que permitirá que o controlador possa direcionar, aumentar ou diminuir a velocidade da aeronave e controlar sua altitude. Diante desse fato, faz-se necessário o conhecimento dos movimentos do VANT propriamente dito, o que pode ser visto na Figura 10.

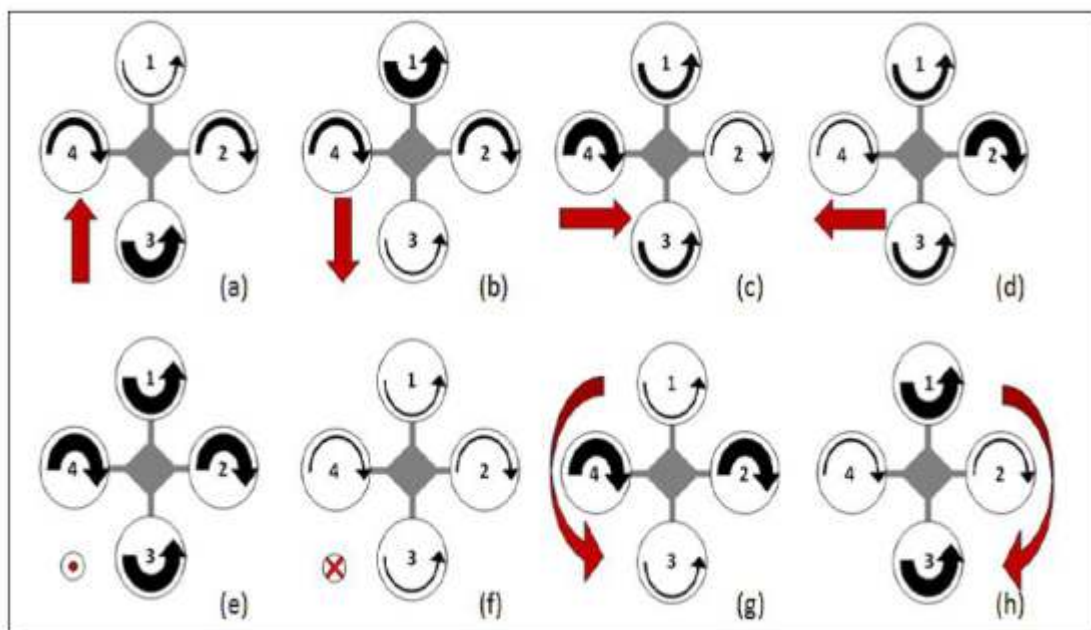


Figura 10 - Rotações de um quadrotor.

FONTE: (SÁ, 2012).

Na figura observa-se que as setas pretas indicam a rotação das hélices e as setas vermelhas representam o resultado do movimento. Além dos quatro movimentos básicos, a aeronave ainda pode se movimentar verticalmente. Para isso, pode aumentar ou diminuir igualmente a velocidade de todos os motores ao mesmo tempo como exposto na Figura 6(e, f). Para girar no sentido horário, os motores 1 e 3 aumentam sua força enquanto os motores 2 e 4 diminuem. Se fizer ao contrário, o quadrotor gira no sentido anti-horário (SÁ, 2012).

A implementação de um sistema de controle para um quadrotor pode ser menos complexo em relação a outros modelos de VANTs. Segundo Guimarães (2012), esse sistema possui 4 variáveis de entrada que se referem aos quatro motores, e seis variáveis de saída, que são x, y, z, roll, pitch e yaw, e estão relacionadas aos ângulos de rotação, conforme Quadro 4.

Quadro 4 - Variáveis de movimentos dos VANTS

QUADRO IV		
VARIÁVEIS	DESCRIÇÃO TÉCNICA	DESCRIÇÃO POPULAR
Roll	Nome dado ao movimento em torno do eixo horizontal, na direção do eixo longitudinal.	Rolagem
Pitch	Nome dado ao movimento em torno do eixo horizontal, perpendicular ao eixo longitudinal	"Levantar o nariz" ou arfagem
yaw	Nome dado ao movimento em torno do eixo vertical, perpendicular ao eixo longitudinal	Guinada

Pensar no projeto que controle todas essas variáveis aliadas ao acoplamento de itens volúveis e as linearidades referentes à aerodinâmica, acionamentos e muitas outras diversidades de detalhes, é uma tarefa extremamente complicada. Uma estratégia para entender o modelo do sistema é apresentada como alternativa para driblar as dificuldades (MICHAEL et al., 2010).

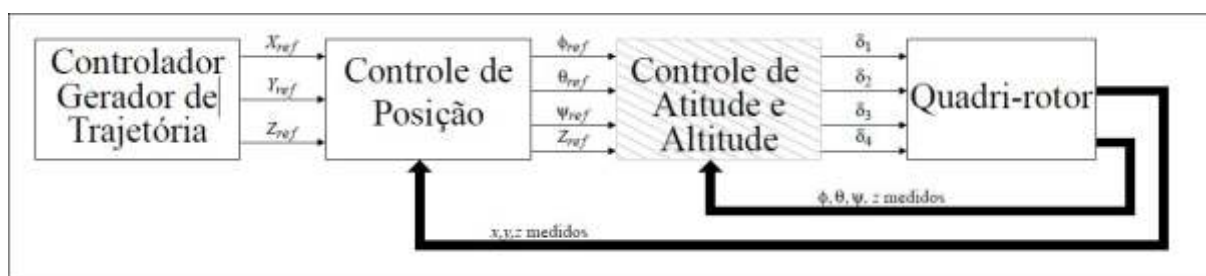


Figura 11 - Estratégia para controle de um quadrotor.

FONTE: (GUIMARAES, 2012).

Guimarães (2012) explica que a malha interna na Figura (controle de atitude e altitude) estabiliza o sistema em uma determinada orientação e altura de referência. O controlador gera os sinais de entrada para cada um dos motores com base nos ângulos de orientação. A malha externa tem a função de gerar referências angular e de altitude para a malha interna. Isso possibilita a navegação com o objetivo de posicionar o veículo em um ponto x, y, z .

Toda a problemática com as variáveis diversas para o controle e navegação de um VANT pode ser solucionado com a agregação da biblioteca MAVLink ao projeto de implementação. Trata-se de uma biblioteca especial de comunicação desenvolvida para aeronaves não tripuladas, ou tripuladas remotamente.

Para a implementação da interface com o usuário foi utilizado o modelo de controle Joystick, com base na biblioteca Joystick da mobile-anarchy-widgets. O primeiro problema resolvido foi a distancia entre os controles, independente do tipo de dispositivo que o usuário venha usar, depois, quais botões adicionais seriam necessários para o caso de navegação do VANT. A tela da interface do usuário é apresentada na Figura 12.

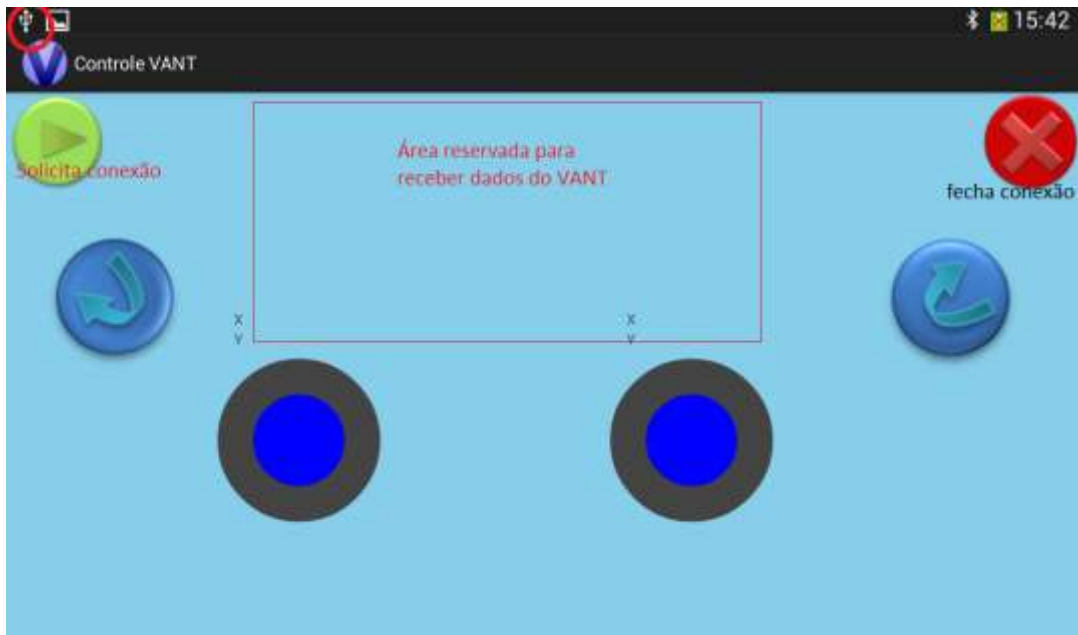


Figura 12 - Tela para o controle do VANT

Cada controle Joystick tem liberdade de até 8 movimentos, o que formula até 16 no caso dos dois joysticks. Para o VANT são necessários apenas 8 movimentos essenciais. A Figura 13 mostra como pode ser o utilizado o joystick para o VANT de forma que atenda a todos os movimentos.

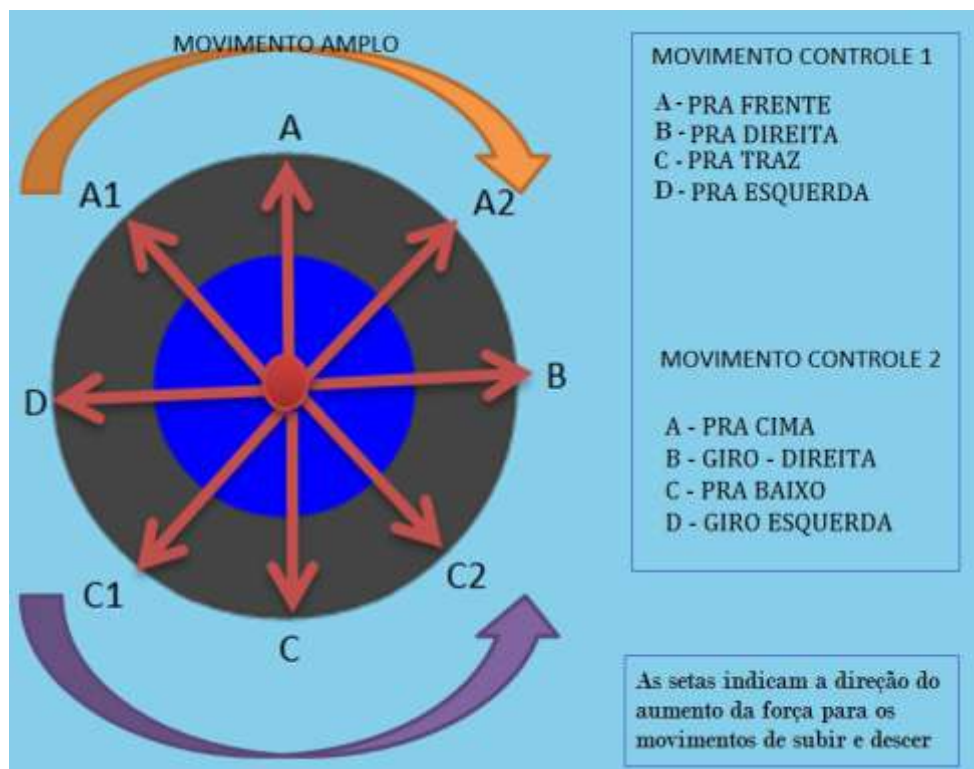


Figura 13 - Configuração do Controle Joystick

As letras A, B, C e D são variáveis implementadas nos códigos tanto para o Arduino como no software para Android. Dessa forma a cada movimento do controle, esse valor é enviado para o Arduino. A parte do código em Java para Android que trata essas variáveis pode ser visto na Figura 14.

```

@Override
public void onNewData(final byte[] data) {
    SerialConsoleActivity.this.runOnUiThread(new Runnable() {
        @Override
        public void run() {
            try {
                SerialConsoleActivity.this.updateReceivedData(data);
            } catch (IOException e) {e.printStackTrace();}
        }
    });
}
}

private JoystickMovedListener _listener1 = new JoystickMovedListener() {
    @Override
    public void OnMoved(int pan, int tilt) {
        txtX.setText(Integer.toString(pan));
        txtY.setText(Integer.toString(tilt));
        if(pan == 10 && tilt == 10){
            t1.setText("FRENTE");
            controle = 1;
        }
        else if(pan == -10 && tilt == 10){
            t1.setText("Esquerda");
            controle = 2;
        }
    }
}

private void updateReceivedData(byte[] data) throws IOException {
    data = new byte[1];
    if(controle == 2){
        data[0] = 'D';
    }
    if(controle == 1){
        data[0] = 'C';
    }
    if(controle == 3){
        data[0] = 'B';
    }
    if(controle == 4){
        data[0] = 'A';
    }
}
}

```

"Aqui o joystick recebe a variável de que corresponde ao movimento"

Aqui modifica o valor do Byte que será enviado para o Arduino

vrDriver.write(data, 200);

Escreve na porta Serial com delay de 200 milisegundos. Tempo necessário para interpretação dos valores.

Figura 14 - Parte do código que trata as variáveis do controle joystick

Além dos dados que são enviados por meio do controle até o dispositivo que os recebe, também tem o sentido oposto, em que o VANT envia dados importantes para o dispositivo móvel. Esses dados são referentes à localização, o que é imprescindível para o bom desempenho do controle remoto no sentido de manter a estabilização do aparelho. Para os testes foi realizado o retorno de dados do sensor de temperatura, mas no VANT os dados são bem mais complexos, pois envolve informações de coordenadas locais em tempo real para saber exatamente a altitude, longitude, latitude,

e todos os outros pontos que podem ser programados conforme a utilização do VANT.

Na tela principal foi disponibilizado campo reservado para dados a ser implementado, e no caso das informações de localização, é necessário utilizar a tela no sentido vertical, para maior comodidade do usuário e maior visibilidade na tela, pois o modo de estabilização utiliza a média de 70% do espaço da tela. Esse modelo de tela é apresentado na Figura 15.

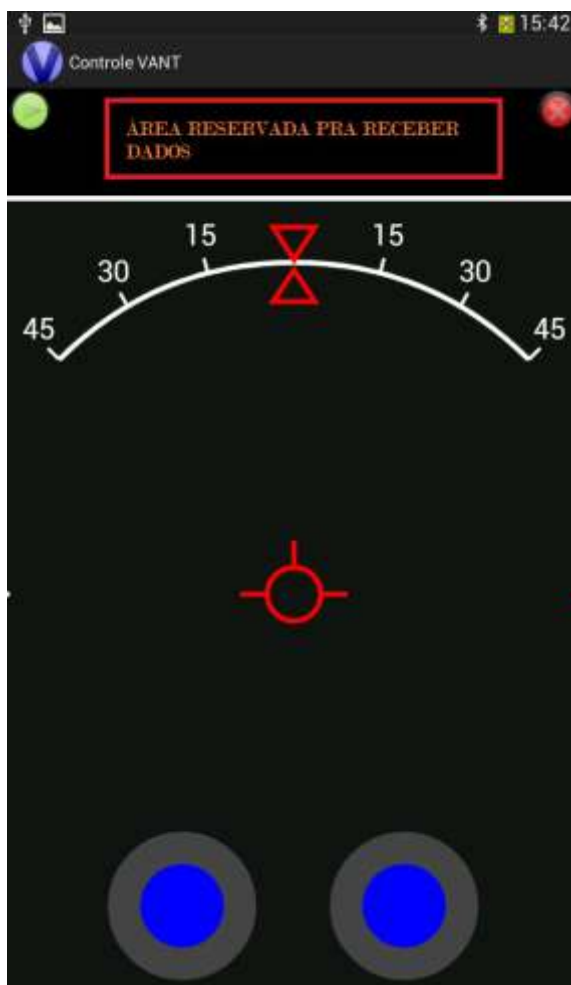


Figura 15 - Tela com modelo de estabilização

Nota-se que a parte reservada para receber dados está exposta no modo vertical também, porém o espaço é bem menor em relação ao anterior. Então, se o usuário quiser receber imagem da localização ao invés de coordenadas, por exemplo, a tela horizontal é a mais indicada. Para o desenho dessa tela foi implementado o modo HUD-Head Up Display, que tem por objetivo manter a foco do usuário no movimento da aeronave representado na tela juntamente com informações de localização, velocidade. e estabilização.

CONSIDERAÇÕES FINAIS

O presente trabalho envolveu desenvolvimento para mobile e implementação para hardware, no caso dos VANTS e permite prática dos assuntos teóricos e conceitos amplos de diversos ramos de ciências. Entre os assuntos trabalhados destacam-se:

1. Protocolos de comunicação de rede - O padrão IEEE 802.11 b/g, é o protocolo necessário para a comunicação de rede sem fio entre a plataforma Arduino e o sistema Android;

2. Comunicação de rede ponto a ponto ;

3. Integração de plataformas - A integração entre a tecnologia Android e o sistema do VANT foi o ponto central do estudo;

4. Programação para dispositivos móveis - O conhecimento adquirido em aulas de programação foi abstraído para o projeto;

5. Programação para Arduino - As atividades de arquitetura de computadores com a prática em Arduino favoreceu a execução da criação dos protótipos e testes;

6. Administração de projetos - Tema importante e indispensável para o desenvolvimento do sistema.

Os testes de comunicação entre o dispositivo móvel e a plataforma Arduino trouxeram uma perspectiva positiva para que futuras aplicações possam ser trabalhadas com a utilização da arquitetura Arduino e comunicação sem fio. Todo o estudo teórico realizado em torno de criar uma conexão entre os dois aparelhos foi contemplado na aplicação.

Apesar de otimista com os resultados meramente em ambientes de simulação, já que ainda não foram possíveis testes reais com o VANT devido resguardo dos equipamentos necessários, almeja-se em um futuro próximo o controle com o VANT em área aberta para testes reais.

REFERÊNCIAS

ANDROID DEVELOPERS - Documentação . Disponível em:

<<http://developer.android.com/reference/android/hardware/usb/package-summary.html>> Acesso em: 24 de janeiro de 2014.

BENEMANN, A. **Estação De Controle Para Veículos Aéreos Não Tripulados** – Universidade do Rio Grande do Sul – Escola de Engenharia – Departamento de Engenharia – Porto alegre, 2013.

BRASSCOM - Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação - **Mobilidade, Inteligência de Mercado**. Periódico. Disponível em:<

<http://www.brasscom.org.br/brasscom/Portugues/download.php?cod=633>> Acesso em: 20 de julho de 2014.

BRESSAN, M. B. **Introdução à Plataforma Arduino** - Minicurso. IFTO - Instituto Federal de Educação, Ciências e Tecnologia. Sudeste de Minas Gerais. Campus de Juiz de Fora - MG, 2014.

GNU - Lesser General Public License. Disponível em:

< <http://www.gnu.org/licenses/lgpl.html> > Acesso em: 20 de julho de 2014.

GUIMARÃES, J. P. F., **Controle de Atitude e Altitude Para Um Veículo Aéreo Não Tripulado Do Tipo Quadrotor**. Universidade Federal do Rio Grande do Norte. Programa de Pós Graduação em engenharia elétrica e de computação. 2012.

MICHAEL, N., MELLINGER, D., LINDSEY, Q. & KUMAR, V. (2010), The grasp multiple micro-uav testbed, Robotics Automation Magazine, IEEE 17(3), 56 –65.

OLIVEIRA, F, A., **CTA e o Projeto VANT**. In: 1º Seminário Internacional de VANT. São José dos Campos, 2005. Palestra proferida no Centro Tecnológico da Aeronáutica em 11 jun. 2005.

SÁ, R. C. **Construção, modelagem dinâmica e controle Pid para estabilidade de um veículo aéreo não tripulado do tipo quadrotor**. Universidade Federal do Ceará, Fortaleza, 2012.

SENAI - Serviço Nacional de Aprendizagem Industrial. **Fundamentos de Rádio Frequência**. Curso de Redes sem Fio. Disponível em:

<<http://pt.slideshare.net/carlosvmelo/fundamentos-de-radio-freqncia>> acesso em: 10 de janeiro de 2014.

SILVA, F. **Comunicação Serial - Arduino e Game Maker** - Tutorial. Disponível em: <<https://www.ufmg.br/frmf/wcontent/uploads/2012/07/TutorialComunica%C3%A7%C3%A3o-Arduino-e-Game-Maker1.pdf> > Acesso em: 23 de julho de 2014.

STUFF, O . **Entenda a escrita em uma porta lógica digital**. Disponível em:< <http://odelotstuff.files.wordpress.com/2013/12/2-1.png> > acesso em 19 de junho de 2014.

VASCONCELOS, C,S. **Projeto, Construção e Controle de quadrotor**. UFRJ - Departamento de Eletrônica e Computação, 2013.