



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

CAIO LUIZ GOMES DE SOUSA ALVES

**HELICÓPTERO CONTROLADO VIA COMPUTADOR COM UMA
ESTAÇÃO DE MEDIÇÃO**

Campina Grande, Paraíba
Abril de 2014

CAIO LUIZ GOMES DE SOUSA ALVES

HELICÓPTERO CONTROLADO VIA COMPUTADOR COM UMA ESTAÇÃO DE MEDIÇÃO

*Trabalho de Conclusão de Curso submetido à
Unidade Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.*

Área de Concentração: Instrumentação Eletrônica

Orientador:

Professor Jaidilson Jó da Silva, D. Sc.

Campina Grande, Paraíba
Abril de 2014

CAIO LUIZ GOMES DE SOUSA ALVES

HELICÓPTERO CONTROLADO VIA COMPUTADOR COM UMA ESTAÇÃO DE MEDIÇÃO

Trabalho de Conclusão de Curso submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Instrumentação Eletrônica

Aprovado em / /

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Professor Jaidilson Jó da Silva, D. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

Dedico este trabalho este trabalho aos meus pais, Guilherme e Fátima, a quem dedico todo meu amor e minha gratidão.

RESUMO

Veículos aéreos não tripulados estão, atualmente, em grande utilização, fazendo o patrulhamento do espaço aéreo em busca de possíveis hostilidades, como na busca por vítimas em desastres. Mas eles não se limitam ao campo militar, seu uso está em grande expansão no campo de pesquisas relacionadas ao melhoramento dos plantios das culturas, na análise climática do ambiente e do ar que tal cultura está sendo cultivada. Para auxiliar os agroempresários e os profissionais que necessitam analisar as condições climáticas, como meteorologistas, então foi proposto um sistema de baixo custo, com o gasto total aproximadamente de R\$ 400,00, de um helicóptero de rádio controle autônomo com sensor de temperatura e umidade, no veículo estaria inserido um *Arduino*, um GPS, um acelerômetro e um comunicador sem fio visando realizar o controle do mesmo e a autonomia, fazendo ainda o envio, em tempo real, dos dados que são coletados pelo helicóptero e o sensor.

LISTA DE FIGURAS

Figura 1: Representação VANT Asa fixa (Hermes 450 FAB).....	2
Figura 2: representação vant asa rotativa (Helicóptero).....	2
Figura 3: Representação VANT asa rotativa (quadricóptero).....	2
Figura 4: Representação vant asa rotativa (helicóptero coaxial).....	3
Figura 5: Fluxograma do Sistema.....	5
Figura 6: Representação do Uso do Servo motor no ângulo de <i>Pitch</i>	6
Figura 7: Representação de um Helicóptero RC Coaxial.....	6
Figura 8: Representação do modelo do helicóptero a ser usado.....	7
Figura 9: Fotografia do Microcontrolador PIC.....	8
Figura 10: Fotografia do Arduino nano.....	9
Figura 11: Representação do funcionamento do GPS.....	10
Figura 12: Representação do GPS.....	11
Figura 13: Fotografia do Módulo GPS + PCI.....	12
Figura 14: Representação da comunicação sem fio.....	12
Figura 15: Fotografia do APC220.....	14
Figura 16: Representação do Programa de configuração de Rede.....	14
Figura 17: Representação do Funcionamento do acelerometro piezoelétrico.....	15
Figura 18: Representação do funcionamento do acelerômetro piezoresistivo.....	16
Figura 19: Representação do funcionamento do acelerômetro capacitivo.....	16
Figura 20: Representação do Acelerômetro ADXL345.....	17
Figura 21: Representação do sensor DHT11.....	18
Figura 22: Esquemático da montagem do sistema.....	20
Figura 23: Representação do conceito de distância entre dois pontos.....	23
Figura 24: Representação dos Pontos no plano cartesiano.....	23
Figura 25: Tela principal do <i>software</i>	25
Figura 26: Representação da tela de visualização dos dados salvos.....	26

SUMÁRIO

1	Introdução	1
1.1	Objetivos	3
2	Componentes do Sistema.....	5
2.1	Helicóptero de Rádio Controle	5
2.2	Microcontrolador	7
2.3	Módulo GPS	9
2.4	Módulo de Comunicação Sem Fio.....	12
2.5	Acelerômetro	14
2.6	Sensores Umidade e Temperatura	17
3	Resultados Obtidos	20
3.1	Protocolo de Comunicação	21
3.2	Determinação da Trajetória.....	23
3.2	Estação de Base	24
4	Considerações finais	26
	Referências	28
	Anexo A.....	30
	Anexo B.....	31
	Anexo C.....	36
	Anexo D.....	37

1 INTRODUÇÃO

Para a realização deste trabalho, será utilizado um veículo classificado como asas rotativas, o helicóptero, pelo fato que será necessário a utilização de alguns dos seus atributos, como a decolagem e pouso vertical e o voo pairado, ao qual, o veículo permanece sobrevoando um local fixo.

A utilização de helicópteros não-tripulados tem sido vasta, em muitas áreas da indústria, da pesquisa e do resgate. Em 2005, o grupo de pesquisa CRASAR (Center for Robot Assisted Search And Rescue), da University of South Florida, nos Estados Unidos da América, empregou um helicóptero não-tripulado na busca por sobreviventes e para avaliar danos causados pela passagem do furacão Katrina pela cidade de Nova Orleans, o referido veículo estava equipado com diversas câmeras (Ko; Lau, 2009).

Na indústria petrolífera, esses veículos são empregados no monitoramento de dutos que transportam petróleo e dos seus tanques de armazenamos, para detecção de falhas ou rachaduras em suas estruturas, evitando a ocorrência de acidentes de maiores proporções.

Os veículos aéreos não tripulados (VANT's), conhecidos como drones, foram criados para melhorar as condições de vida dos seres humanos e dar segurança as pessoas que trabalham na execução de tarefa, cujo grau de periculosidade é alto. Normalmente, são empregados na realização de missões militares, mas não sendo exclusivamente utilizados nesta área, sendo também alocados em outros segmentos da indústria, como a agrícola, petrolífera, energética, dentre outras que vislumbam aplicações usando as características dos VANT's.

Seisan (2012) classifica os VANT's em relação a movimentação ou não de suas asas. Os veículos de asas fixas, como é o caso do avião (figura 1), é ideal para operações onde o deslocamento tenha que ser feito em um curto espaço de tempo. Já os de asas rotativas, como o helicóptero (figura 2), quadricóptero (figura 3) e helicóptero coaxial (figura 4), são ideais para aplicações que requer um voo pairado ou decolagem e pouso vertical.



FIGURA 1: REPRESENTAÇÃO VANT ASA FIXA (HERMES 450 FAB)



FIGURA 2: REPRESENTAÇÃO VANT ASA ROTATIVA (HELICÓPTERO)



FIGURA 3: REPRESENTAÇÃO VANT ASA ROTATIVA (QUADRICÓPTERO)



FIGURA 4: REPRESENTAÇÃO VANT ASA ROTATIVA (HELICÓPTERO COAXIAL)

Do ponto de vista científico, como mencionado por Pestún, et. al (2011), o helicóptero é apresentado como um sistema instável, de múltiplas entradas e múltiplas saídas - MIMO, e não linear, portanto, ele se torna um sistema complexo para se tornar estável.

A adição de um par de hélices girando em sentido contrário ao outro par de hélices, é uma das práticas aplicadas por muitos projetistas de helicópteros para tentar torná-lo menos instável e menos complexo, conseqüentemente. Essas hélices, usualmente, tendem a girar em velocidades distintas, possibilitando o aparelho não fazer movimentos rotacionais em sua carcaça. Com esta prática, o helicóptero passa a ter a nomenclatura de coaxial.

1.1 OBJETIVOS

Visando ajudar os produtores agrícolas, meteorologistas e outros profissionais que necessitem dos dados das variações climáticas, este trabalho tem como objetivo a instrumentação de um veículo não tripulado para realizar a análises do ar em um determinado local, proporcionando a tomada de algumas decisões, como o período de irrigação e quais as melhores épocas para o plantio da cultura. Com os dados de temperatura e umidade coletados, eles também poderão tomar conhecimento do clima, prevendo grandes períodos de chuva ou de seca.

A implementação deste trabalho será feito com o uso de um helicóptero coaxial, tendo acoplado a ele os seguintes componentes: um módulo GPS, um microcontrolador, um módulo de transmissão sem fio e uma estação móvel de medição meteorológica. Ao

qual, o veículo realize uma determinada trajetória, selecionada através de coordenadas.

2 COMPONENTES DO SISTEMA

O sistema será implementado de acordo com o fluxograma apresentado na figura 5:

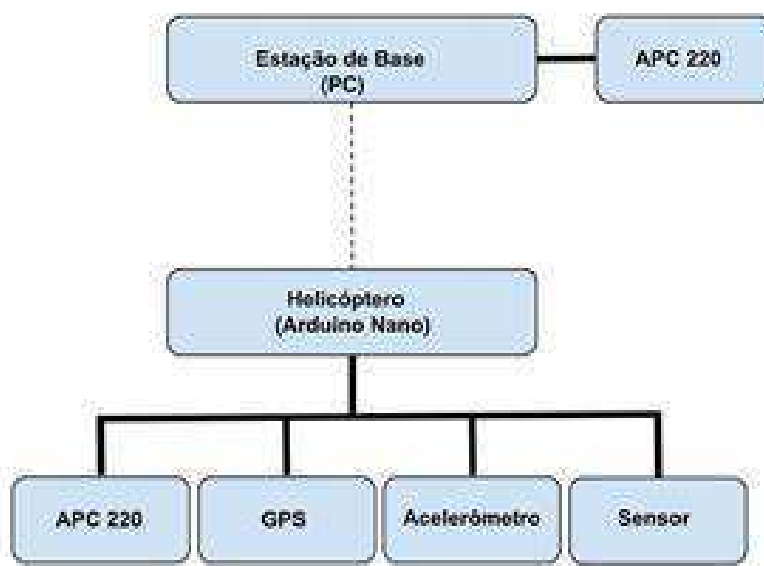


FIGURA 5: FLUXOGRAMA DO SISTEMA

2.1 HELICÓPTERO DE RÁDIO CONTROLE

Segundo Tay, Chua e Iqbal (2012), apesar dos helicópteros de controle remoto serem tido como brinquedos, eles são bem projetados, possibilitando que usuários menos experientes possam manuseá-los de forma mais estável, tendo menos riscos de colisão.

De acordo com Cha (2008), eles são classificados de acordo com a quantidade de canais de comunicação, podendo ser: 2, 3, 4 e 6 canais. Os de 2 e 3 canais, apresentam mobilidade limitada, apresentando os movimentos de subida, descida, direita e esquerda, sendo considerado para usuários iniciantes. Os helicópteros de 4 e 6 canais se diferenciam pela possibilidade de alteração dos ângulo de *pitch*, que é o

ângulo das hélices em relação ao eixo horizontal, ao qual essa alteração é feita pelo acionamento de servo motores, apresentado na figura 6.



FIGURA 6: REPRESENTAÇÃO DO USO DO SERVO MOTOR NO ÂNGULO DE *PITCH*

Segundo Mishra (2013), para compensar que a fuselagem do helicóptero gire em sentido contrário ao da rotação das hélices, então os há a adição de um par de hélices que giram em sentidos opostos ao par de hélices principal fazendo com que a fuselagem do helicóptero não gire em torno do seu próprio eixo, como é apresentado na figura 7.



FIGURA 7: REPRESENTAÇÃO DE UM HELICÓPTERO RC COAXIAL

Com os helicópteros RC de até 4 canais e coaxial, normalmente, os rotores de calda estão voltados para cima, como pode ser visto também na figura 3. Com isso, ele tem a função de guiar o helicóptero para frente ou para trás, dependendo de como é acionado.

Para este projeto, foi então escolhido um helicóptero de 4 canais e coaxial, que sua representação pode ser vista na figura 8.



FIGURA 8: REPRESENTAÇÃO DO MODELO DO HELICÓPTERO A SER USADO

De acordo com o manual do fabricante, ele apresenta as seguintes características:

- Autonomia de Voo: 10 a 20 minutos
- Altitude máxima: 20 metros
- Frequência do rádio comunicador: 27MHz

2.2 MICROCONTROLADOR

Segundo Denardin (2008), o microcontrolador é um sistema computacional completo que em sua estrutura interna, estão incluído elementos, tais como CPU; memórias de armazenamento de dados, e somente de leitura para o armazenamento de programas; a memória EEPROM; um sistema de clock, para dá provimento as atividades da CPU; portas de entrada e saída para uso geral; e dispositivos periféricos, tais como conversores analógico/digital (AD) e digital/analógico (DA), tendo dois tipos de arquitetura, a RISC e a CISC.

De acordo com Rodrigues (2011), a arquitetura RISC se caracteriza pela redução do tempo médio de execução das instruções de máquinas, com isso, o número de ciclos de clock também diminui, mas a quantidade de instruções que é executada por programa aumenta, então a maioria das instruções são executadas em um único ciclo de clock. Já a arquitetura CISC, apresenta instruções mais complexas com a intenção de diminuir o número de instruções que um programa necessita, mas o número de ciclos de clock por instruções pode aumentar, tendo em vista a complexidade da execução de determinada

instrução, então essa arquitetura é caracterizada como sendo mais lenta a execução das suas instruções.

Apesar de realizar as mesmas funções específicas que um microprocessador, apresentando os componentes lógicos e aritméticos, os microcontroladores se diferenciam pelo fato de que neles todos os elementos mencionados anteriormente já estão integrados no mesmo circuito integrado, já no microprocessador há a necessidade da conexão de algum periférico. Uma representação de um microcontrolador pode ser visto na figura 9.

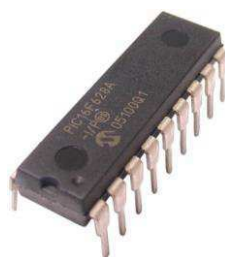


FIGURA 9: FOTOGRAFIA DO MICROCONTROLADOR PIC

Como afirma Denardin (2008), devido ao alto custo dos dispositivos eletrônicos nos anos anteriores, o uso de microcontroladores limitou-se ao eletrodomésticos com alta tecnologia, o caso da televisão, vídeo, som. Mas com a caída nos preços dos circuitos integrados, então eles passaram a ser utilizados em outros produtos que não tinham uma tecnologia tão avançada, isso possibilitou o desenvolvimento de produtos que não existiam, como os aparelhos biomédicos, detectores de radar, dentre outros, fazendo com que aumentasse também o conforto e a segurança dos indivíduos.

Segundo Oki e Mantovani (2013), os principais fabricantes são: Intel, Microchip, Freescale (Motorola), Texas Instruments e Atmel. A maioria dos fabricantes apresentam dispositivos com variados tamanhos de barramento, desde 8-bits até 32-bits, e também com os dois tipos de arquiteturas: RISC e CISC.

Para este trabalho, será usado um microcontrolador da fabricante Atmel, de 8-bit e com sua arquitetura baseada na RISC, o ATmega328.

Para realizar o trabalho com esse microcontrolador, será usado uma placa de prototipagem eletrônica, Arduino, desenvolvida na Itália, para que possa desempenhar as funções de forma mais simples, direta e com uma proposta *open-source*.

Apesar de ter inúmeras placas Arduino, será utilizada a Arduino nano, que pode ser vista na figura 10, por apresentar um design mais enxuto, ser mais portátil e leve. Com isso, não haverá tamanha interferência do peso no helicóptero.



FIGURA 10: FOTOGRAFIA DO ARDUINO NANO

De acordo com o site da plataforma Arduino, a placa apresenta as seguintes características:

- Tensão de Operação: 5V
- Tensão de Alimentação: 7 – 12V
- Pinos digitais de entrada/saída: 14 (6 podem prover saída PWM)
- Entrada Analógica: 8
- Memória Flash: 32KB
- EEPROM: 1KB
- Frequência do clock: 16MHz
- Comunicações: I2C, Serial, SPI

Com ela, será feita a aquisição dos dados dos outros dispositivos, como GPS, Acelerômetro, módulo de comunicação e sensores de temperatura e umidade. Também será feito o acionamento dos motores do helicóptero e o envio, pela porta Serial, dos dados do sensor coletados, para a estação de base.

2.3 MÓDULO GPS

Dispositivo presente na maioria dos veículos autônomos, utilizado para estabelecer sua localização global.

Como mencionado por Alves (2004), o GPS foi desenvolvido pelo Departamento de Defesa dos Estados Unidos da América. O projeto se iniciou no ano de 1973, com a finalidade de ajudar as forças armadas, com a determinação do posicionamento das tropas e das aeronaves, bem como, proteger o país de possíveis ataques de seus inimigos, pois o mesmo estava em conflito em diversas localidades do mundo, estava no auge da Guerra Fria. Seu uso civil foi liberado pelo governo

americano de forma mais restrita, apresentando menor precisão, sendo utilizado através do serviço *Standard Positioning Service (SPS)*.

Ainda segundo Alves (2006), o sistema é composto por 24 satélites varrendo seis órbitas estacionárias, para que cada órbita seja trafegada por 4 satélites, os quais completam tal trajetória em 12 horas e tem 28° de visualização, como pode ser visto na figura 11. E todos eles são controlados por estações bases que ficam espalhadas pela superfície terrestre, tendo uma delas considera a estação mestra que fica localizada no Colorado (EUA).

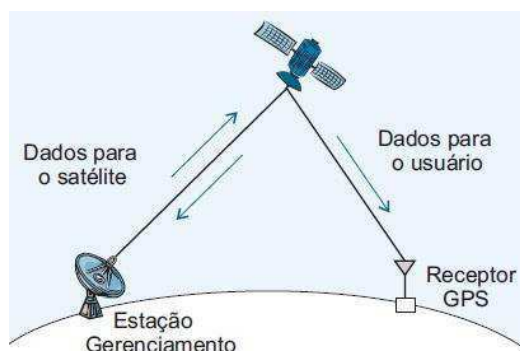


FIGURA 11: REPRESENTAÇÃO DO FUNCIONAMENTO DO GPS

A determinação do posicionamento de certo objeto pode ser feita através de dois métodos: o pontual e o relativo, sendo o método pontual o mais usual para baixa precisão, uso civil.

No método pontual, para a determinação, o usuário só precisa se dispor de um único receptor, o que difere em relação ao método relativo, o qual há a necessidade de se ter mais receptores, mas com o surgimento dos Sistemas de Controle Ativos (SCA) agora pode-se utilizar o método relativo utilizando apenas um receptor, acessando os dados das estações de base mais próximas.

De acordo com a apostila do Curso de GPS e Cartografia Básica (2012), o satélite emite um sinal para um receptor na superfície terrestre, então o receptor cronometra o tempo de propagação deste sinal e faz uma comparação com o valor do tempo de propagação para a estação base, que é previamente definido no aparelho, com a diferença, na ordem de décimo de segundos, então o sinal recebido é multiplicado pelo valor da velocidade de propagação da luz (aproximadamente $2,9979 \times 10^8$ m/s), com isso, tem-se a localização de um determinado objeto sobre uma superfície imaginária esférica.

Como mencionado anteriormente, 4 satélites abrangem cada uma das 6 órbitas terrestres, e todos emitem sinal para o receptor do objeto a ser localizado, fazendo uma

triagem de menor tempo de propagação e com isso, tem-se a distância do objeto, sendo que a localização não é fornecida coordenadas cartesianas, mas são usadas coordenadas geográficas, como latitude, longitude e elevação.

Com o intuito de usar um módulo GPS de pequenas dimensões e consequentemente de alta performance na recepção dos sinais, foi adquirido o *NEO-6 GPS Modules* (figura 12) que possibilita uma grande quantidade de conexões, e com seus 50 canais, que emprega o mecanismo *Time-To-First-Fix (TTFF)*.



FIGURA 12: REPRESENTAÇÃO DO GPS

Este módulo apresenta as seguintes características:

- Compatível com as interfaces: UART, USB, SPI e DDC;
- Sensibilidade entre -161 a -147 dBm;
- Máxima taxa de atualização de navegação: 5Hz;
- Limite de altitude de operação: 50.000 m;
- Limite da velocidade de operação: 500m/s.

Para ser mais fácil de fazer a conexão com o microcontrolador, então esse módulo foi adicionado a uma placa de circuito impresso com reguladores de tensão, para baixar a tensão de 5 V para 3,3 V, um conversor das saídas do módulo para uma saída compatível com o microcontrolador, nesse caso uma saída serial, e a conexão com uma antena para aumentar a potência do sinal. Pode ser vista uma representação da placa na figura 13.



FIGURA 13: FOTOGRAFIA DO MÓDULO GPS + PCI

2.4 MÓDULO DE COMUNICAÇÃO SEM FIO

Concede mobilidade aos dispositivos, enviando sinais através de ondas de rádio evitando a utilização de fios, podendo ser usado desde simples comunicadores de voz até por transmissão via satélite. A representação do funcionamento pode ser visto na figura 14.

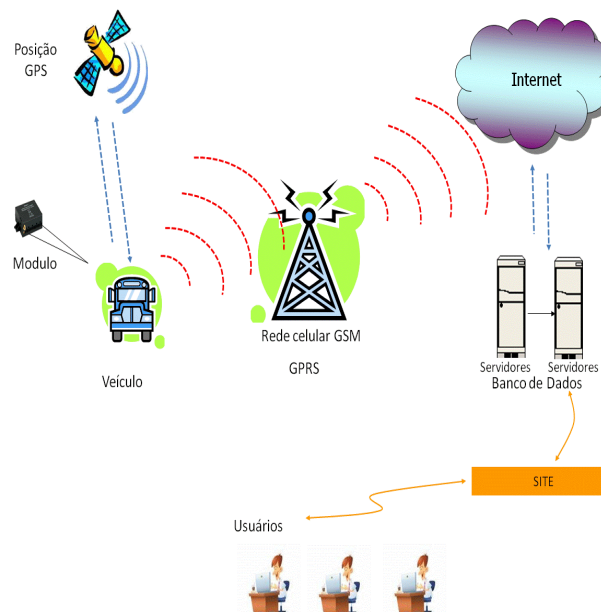


FIGURA 14: REPRESENTAÇÃO DA COMUNICAÇÃO SEM FIO

Com o surgimento do telefone, os indivíduos passaram a ter a necessidade de se comunicar com os outros que estavam em locais mais distantes, mas para interligar áreas longínquas o custo para implementação se tornaria totalmente elevado, baixando a margem de lucro dos empresários do setor. Então, com o avanço nos estudos das ondas

eletromagnéticas, pôde-se interligar áreas muito distantes usando dispositivos que propagassem essas ondas.

Segundo Silva (2008), hoje em dia, a comunicação sem fio é difundida em todos os meios, sendo mais empregada em rede de computadores, ao qual os indivíduos se conectam à internet em determinados locais.

De acordo com Rocker (2009), a comunicação sem fio apresenta 4 tipos, que são: infravermelho, ondas de rádio, micro-ondas e ondas de satélite. Cada uma tem sua característica própria que será mencionada:

- *Infravermelho*: Segundo Nério e Rodrigues (2003), a faixa de frequência é de 300 GHz até, aproximadamente, 429 THz. Os aparelhos tem que está muito próximos, quase colados, não pode ter obstrução física no espaço por onde o sinal vai fluir, e para completar, a taxa de transmissão é muito baixa, não chegando a 10Mbps. Essa tecnologia foi muito empregada na transferência de dados entre celulares.
- *Ondas de rádio*: Segundo Nério e Rodrigues (2003), a faixa de frequência é de 300 Hz até 3 GHz. Mais usadas nas transmissões de rádio, há a necessidade de implantar antenas, mesmo conseguindo atravessar paredes, apresentam baixa capacidade na velocidade de transmissão, podendo ser dividida em: bluetooth, IEEE 802.11 e HomeRF.
- *Micro-ondas*: Conforme Nério e Rodrigues (2003), tem faixa de frequência de 3 GHz até 300 GHz. Utilizadas na comunicação móvel, e necessita de um espaço sem obstrução para serem propagadas.
- *Ondas de satélites*: De acordo com Rocker (2009), apresentam largura de banda muito elevada (500MHz), sendo utilizada para transferência de dados entre grande distâncias, como intercontinental, para esse tipo de comunicação há a necessidade de antenas parabólicas e satélites em órbitas geoestacionárias.

No nosso caso, utiliza-se o APC220, apresentado na figura 15, da empresa Silicon Laboratories, dispositivo que transfere dados através de ondas de rádio, que apresenta grande versatilidade e é ideal para aplicações em robótica, pois pode-se controlar a rede de comunicação com o uso de um *software*, apresentado na figura 16.

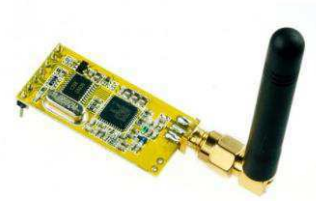


FIGURA 15: FOTOGRAFIA DO APC220

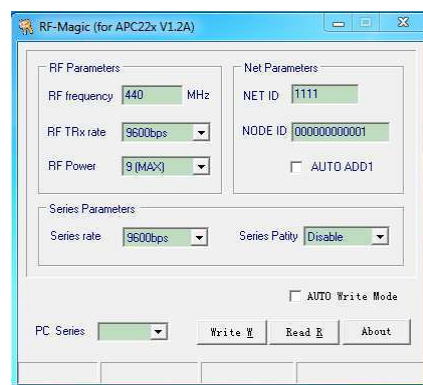


FIGURA 16: REPRESENTAÇÃO DO PROGRAMA DE CONFIGURAÇÃO DE REDE

Além de apresentar interface TTL para se comunicar com microcontroladores, as suas outras principais especificações são:

- Frequência de trabalho: 418 a 455 MHz
- Tensão de Alimentação: 3,3V a 5V
- Abrangência: 1200m em campo aberto
- BaudRate: 1200 a 19200 bps
- Buffer de entrada: 256 bytes

2.5 ACELERÔMETRO

De acordo com Sampaio (2012), o acelerômetro é um componente capaz de medir a aceleração aplicada sobre um corpo, levando em consideração a 1ª lei de Newton, que um corpo só muda seu estado de repouso ou de movimento uniforme em linha reta, quando nele são aplicadas forças que o faça mudar o seu estado inercial.

Segundo Fusco (2010), esses dispositivos são bastante utilizados em diversos segmentos tanto da indústria como na medicina. Algumas das aplicações está relacionado ao uso deles em carros para o controle do *air-bags*, nos modernos aparelhos celulares, *smartphones*, que são utilizados para detectar a posição da tela, são utilizados também para proteger os HD's de possíveis impactos que os *notebooks* venham a sofrer. No campo da medicina, podem ser utilizados para caracterizar movimentos, vibrações, na contração muscular, visando facilitar a vida de idosos e pessoas com deficiência física.

De acordo com Rocha e Marranghello (2013), os acelerômetros podem ser divididos em três tipos: piezoelétricos, piezoresistivo e capacitivo. Os acelerômetros piezoelétricos (figura 17) utilizam materiais com a propriedade de gerar uma tensão elétrica ao serem comprimidos, sendo esta proporcional à compressão. Então, eles são ideal quando submetidos a força de tração e de compressão. Nas aplicações que necessitam de medição dos três eixos do plano cartesiano (X, Y, Z), será necessário a utilização de três desses dispositivos em cada eixo a ser medido.

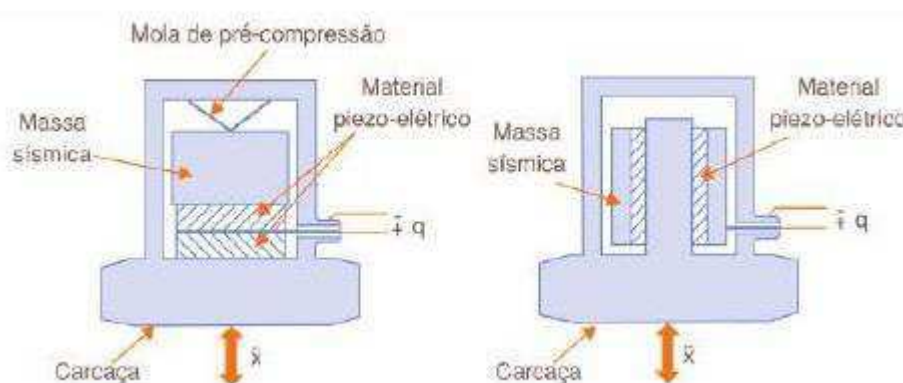


FIGURA 17: REPRESENTAÇÃO DO FUNCIONAMENTO DO ACELEROMETRO PIEZOELÉTRICO

Conforme Silva Filho, Rudiger e Nascimento (2011), os acelerômetros piezoresistivos (figura 18) são bem parecidos com os acelerômetros piezoelétricos apresentando a diferença com relação ao material utilizado, que nesse caso tem a propriedade resistiva, fazendo com que a resistência do material varie quando uma força é exercida. Para esse tipo de acelerômetro, eles são ideais para medir acelerações em valores muito baixo de frequência.

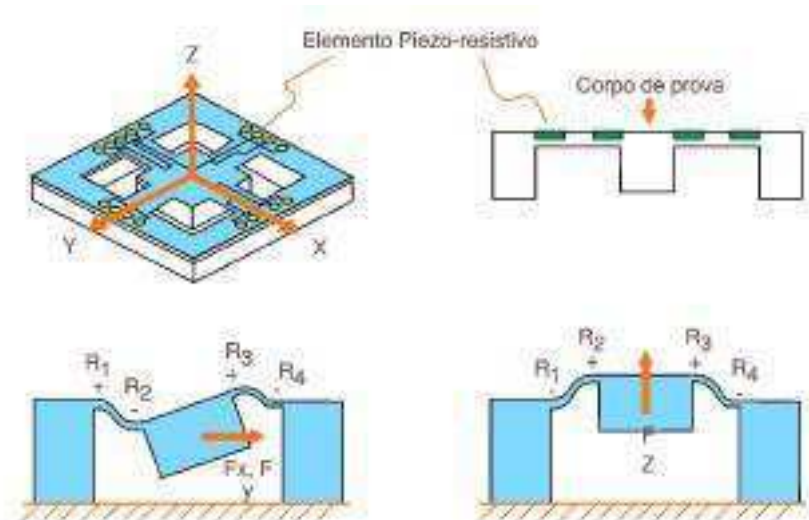


FIGURA 18: REPRESENTAÇÃO DO FUNCIONAMENTO DO ACELERÔMETRO PIEZORESISTIVO

Já o acelerômetro capacitivo (figura 19), segundo Rocha e Marranghello(2013), é o dispositivo mais utilizados no meio comercial, ele utiliza um capacitor para produzir variação de tensão, as placas do capacitor possuem uma diferença de potencial elétrica, ao qual é proporcional à distância entre as placas metálicas, com isso quando uma movimentação ocorre, uma variação de tensão é produzida. Sendo ideal para ambientes que requerem medidas estáveis em baixo nível, baixa frequência. Esse tipo de acelerômetro não precisa da inclusão de mais sensores caso precise fazer medições em outros eixos cartesianos.

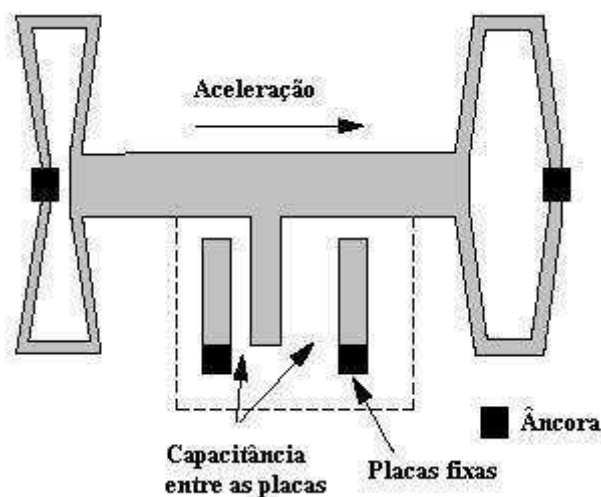


FIGURA 19: REPRESENTAÇÃO DO FUNCIONAMENTO DO ACELERÔMETRO CAPACITIVO

Para este trabalho, foi utilizado um acelerômetro do tipo capacitivo, por apresentar tamanho mais reduzido e realizar medições nos 3 eixos cartesianos (X, Y, Z), seguindo essas características, o acelerômetro escolhido foi o ADXL345, da *Analog Devices*. Ele vem soldado em uma placa, que nela estão inseridos os componentes eletrônicos que são fundamental para um bom funcionamento do acelerômetro. Todos esses componentes eletrônicos juntos exercem duas funções: condicionamento do sinal lido e atenuação da tensão de alimentação do dispositivo. A representação da placa com o acelerômetro ADXL345 é apresentada na figura 20.



FIGURA 20: REPRESENTAÇÃO DO ACELERÔMETRO ADXL345

De acordo com o *Datasheet* do ADXL345, ele apresenta uma alta resolução de medição de até $\pm 16g$, a força da gravidade. O sinal de saída é formado por um conjunto de 16-bit, e ele suporta as interfaces de comunicação SPI e I2C.

2.6 SENSORES UMIDADE E TEMPERATURA

“Um sensor de umidade é um dispositivo transdutor que converte um estímulo físico ou mecânico, causado pela umidade relativa do ambiente, no valor da umidade relativa correspondente” (Castro, 2011).

Ainda segundo Castro (2011), os sensores de umidade podem ser classificados como capacitivo e resistivo. Os sensores do tipo capacitivo são baseados na variação da permissividade dielétrica do material com a umidade relativa do ambiente. Já o sensor de umidade resistivo, quando a umidade relativa apresentar variação a resistência do material que compõem o sensor também será alterada, possibilitando assim a medição da umidade relativa.

De acordo com Shigue (2008), o sensor de temperatura é um transdutor que também converte uma grandeza física, no caso a temperatura, em sinal elétrico. Podem ser classificados por 3 tipos: resistivo, termoeletrico, infravermelho.

Os sensores de temperatura classificados como resistivos tem essa denominação pelo fato da sua resistência variar quando houver uma alteração na temperatura ambiente. Já os termoeletricos, também conhecido como termopares, eles produzem uma tensão quando submetidos ao efeito Seebeck, que é quando dois metais distintos são soldados nas extremidades e uma delas também é aquecida, posteriormente, uma das extremidades é aberta e aparecerá uma tensão nos terminais que foram abertos, com isso, esta tensão está diretamente relacionada com a temperatura.

Ainda segundo Shigue (2008), os sensores de temperatura classificados como infravermelho captam a radiação eletromagnética no comprimento de onda do infravermelho, sendo esta faixa de frequência característica de emissão de calor.

Para este projeto, foi adquirido um sensor em que nele está presente as duas funcionalidades, o sensor de temperatura e o de umidade, que é o sensor DHT11 (figura 21).

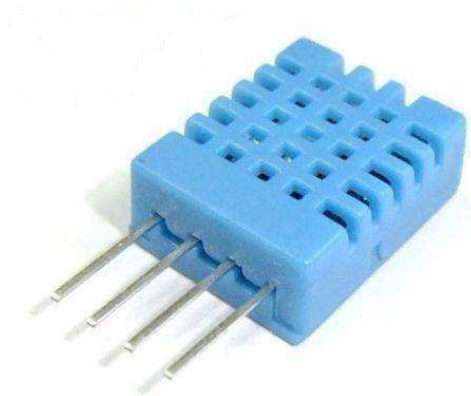


FIGURA 21: REPRESENTAÇÃO DO SENSOR DHT11

Este sensor apresenta os seguintes parâmetros:

- Resolução: 16-bit
- Período de amostragem: mais de 2 segundos
- Faixa de temperatura: 0 a 50 °C
- Faixa de umidade: 20 a 90% RH
- Interface: Digital

A saída é digital, onde os dados são divididos em 8 bits, sendo:

- Umidade: 16 bits, 8 bits MSB e 8 bits LSB
- Temperatura: 16 bits, 8 bits MSB e 8 bits LSB
- Paridade: 8 bits.

Na próxima seção, serão apresentados os resultados obtidos, tais como: o protocolo de comunicação entre o helicóptero de rádio controle e a estação de base, o cálculo da trajetória a ser realizada pelo aparelho, e o programa supervisor que será executado na estação de base, ao qual fará todas as tarefas necessárias para execução do sistema, como envio das coordenadas, recebimento e gravação dos dados.

3 RESULTADOS OBTIDOS

Para realizar a instrumentação do helicóptero de rádio controle, foi integrado à placa de prototipagem Arduino os componentes descritos na seção anterior. A instalação dos componentes foi feita de acordo com a figura 22.

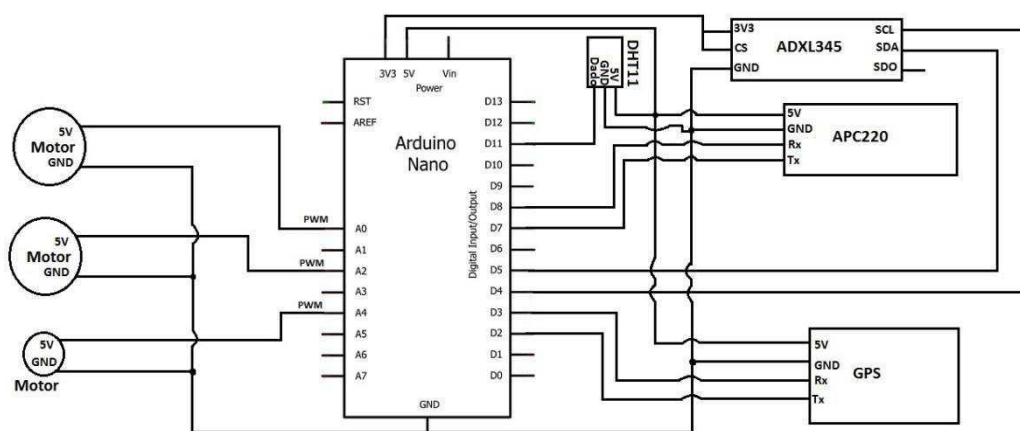


FIGURA 22: ESQUEMÁTICO DA MONTAGEM DO SISTEMA

Como visto na figura 22, o *Arduino* faz a interface, o processamento dos dados que são recebidos pelos dispositivos externos e toma decisão de acordo com o foi programado.

Para que ele possa realizar as funções descritas anteriormente, então foi escrito um código, em uma linguagem de programação adaptada a placa de prototipagem, baseada na linguagem de programação C.

O programa segue a seguinte linha de execução:

- 1) As coordenadas geográficas (latitude e longitude) são recebidas através do GPS;
- 2) Usando o APC220, as coordenadas são enviadas ao computador que está na estação de base;
- 3) O operador escolhe, através do *software* instalado no computador, um valor válido de latitude e de longitude, a ser definido como local de destino;
- 4) O APC220, presente no computador da estação de base, envia as coordenadas escolhidas para o outro APC220, que está no helicóptero;

- 5) Com os dados recebidos, então o *Arduino* realiza o cálculo da distância através do teorema de Pitágoras;
- 6) O *Arduino* aciona os motores do helicóptero, de maneira que a rotação inicial seja mais lenta e vá aumentando até que o veículo decole;
- 7) O acelerômetro, ADXL345, faz a verificação da estabilidade do helicóptero, para que o mesmo não gire em torno do seu próprio eixo;
- 8) Com o helicóptero estável, o GPS monitora as posições geográficas;
- 9) O *Arduino* faz a comparação para vê se está posicionado no sentido para iniciar a trajetória;
- 10) Iniciada a trajetória do helicóptero, a posição geográfica é monitorada ao passo de 100ms, pelo GPS, visando corrigir a trajetória;
- 11) Quando o helicóptero chega ao destino escolhido no item 3, então o sensor de temperatura e umidade (DHT11) começa a fazer a coleta dos dados;
- 12) Os dados do DHT11 são enviados para o computador da estação de base, através do APC220, no são gerados gráficos em tempo real e os dados são armazenados em um arquivo com extensão .txt;
- 13) Através do APC220 da estação da base, então o computador envia um comando para o helicóptero retornar a base, com isso ele cessa a aquisição dos dados do DHT11 e refaz os itens 8, 9, 10.

3.1 PROTOCOLO DE COMUNICAÇÃO

Foi pensado em um protocolo de comunicação para ser usado com os dados das coordenadas, e do sensor de temperatura e umidade tendo em vista, facilitar a compreensão do microcontrolador e da estação de base.

Primeiramente, foi determinado o tamanho da palavra que cada comando teria, para isso foi verificado o tamanho de cada um dos dados que são recebidos pelo microcontrolador, como coordenadas e dados do sensor.

O comando das coordenadas é de sentido bidirecional, pois pode ser enviado da estação de base para o helicóptero, como é feito no instante inicial que o helicóptero ainda está em terra, e o outro sentido é o do helicóptero para a estação de base, afim de monitorar a posição do helicóptero em cada instante.

Verificou-se que o tamanho da palavra da longitude e da latitude tem 6 caracteres, cada uma, sendo os dois primeiros dígitos referentes a dezena e unidade, e os outros 4 aos decimais. Então, para ter um comando menor, foi instituído o acréscimo de uma letra maiúscula no início de cada palavra de comando, sendo empregado a vogal “A” para designar que é a latitude e a vogal “O” para expressar que é a longitude.

A representação de como o comando será passado tanto para o microcontrolador como para o computador é visto a seguir.

$$AxxxxxxOyyyyyy \quad (1)$$

Quando o primeiro caractere é recebido, o programa verifica se é a letra “A”, então, ele entenderá que são as coordenadas e esperará os outros 13 caracteres. Após serem recebidos, eles são processados, separados como sendo latitudes e longitudes, e então passam para a próxima etapa do programa.

O comando do sensor de temperatura e umidade é unidirecional, ou seja, ele só é enviado do microcontrolador para a estação de base, pois é neste comando que estará contido as informações referentes ao valor da temperatura e da umidade que está sendo capturada pelo sensor.

Analogamente feito com o comando das coordenadas, também foi verificado o tamanho do dado que é enviado ao microcontrolador, podemos notar que o tamanho de cada dado é de 3 caracteres, ao qual um caractere será usado como número decimal e os outros dois caracteres como dezena e unidade.

Para manter o padrão do comando das coordenadas, então foi introduzido uma letra em cada início de dados de um determinado tipo, sendo temperatura e umidade. Por critério de escolha própria, foi inicialmente empregado o envio dos dados da temperatura e posteriormente o da umidade. A letra escolhida para representar os dados da temperatura foi o “T” e a letra utilizada para a umidade foi a letra “U”, com isso a representação do comando que é passado do microcontrolador para o computador da estação de base, pode ser visto a seguir.

$$Txx.xUyy.y \quad (2)$$

O programa instalado na estação de base fará a comparação do caractere inicial do comando que é recebido, após isso, ele fará o processamento dos dados e os enviará para cada espaço no programa ao qual são designados.

3.2 DETERMINAÇÃO DA TRAJETÓRIA

Afim de determinar a trajetória mais simples e rápida para o helicóptero foram usados os conceitos básicos da geometria analítica.

Um dos principais conceitos da geometria analítica é quando se quer determinar a distância entre dois pontos, ao qual é enunciado que por dois pontos passa apenas uma reta. A representação pode ser vista na figura 23.

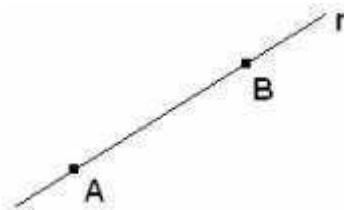


FIGURA 23: REPRESENTAÇÃO DO CONCEITO DE DISTÂNCIA ENTRE DOIS PONTOS

Para calcular a distância entre dois pontos, a melhor forma é colocá-los em um plano cartesiano para se ter as posições (x,y) de cada ponto, como pode ser visto na figura 24.

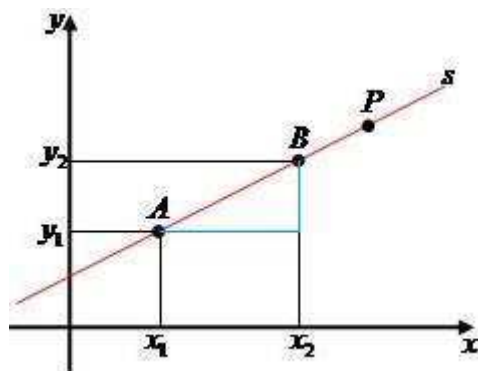


FIGURA 24: REPRESENTAÇÃO DOS PONTOS NO PLANO CARTESIANO

Pode ser visto que a distância forma dois catetos, que são a diferença de cada posição de cada ponto. Um na base, que é a diferença entre o x_1 e o x_2 , dos pontos A e B, e um cateto altura, que é a diferença entre o y_1 e o y_2 , dos pontos A e B, respectivamente.

Então, para o cálculo da distância é utilizado o teorema de Pitágoras, o qual é enunciado da seguinte forma: que a distância entre dois pontos é a hipotenusa do triângulo retângulo formado pelos dois catetos.

A fórmula da distância é expressa da seguinte forma:

$$D^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \quad (3)$$

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4)$$

O helicóptero faz a trajetória baseada na fórmula 4 , apresentando o menor tempo de deslocamento até o ponto desejado.

Mas essa fórmula só é válida dependendo do local ao qual o helicóptero está inserido e pretende se deslocar, tendo que ser um local aberto, sem obstáculos que interfiram na sua trajetória retilínea.

3.2 ESTAÇÃO DE BASE

Na estação de base foi posicionado um computador, nele estava instalado um *software* desenvolvido pelo autor do trabalho.

O *software* foi escrito na linguagem de programação C#, ao qual utiliza ferramentas gráficas para fazer a inserção de dados e a visualização dos mesmos. Posteriormente, com todos os componentes conectados no *layout* do programa, então é feita a lógica de programação utilizando a lógica da linguagem de programação C.

No *software* foi integrado todas as funções que são necessárias a execução do funcionamento do projeto como um todo. Desde a apresentação das coordenadas até o recebimento dos dados referentes ao sensor de temperatura.

A interface do programa pode ser vista na figura 25, ao qual é apresentada a tela principal do *software*.

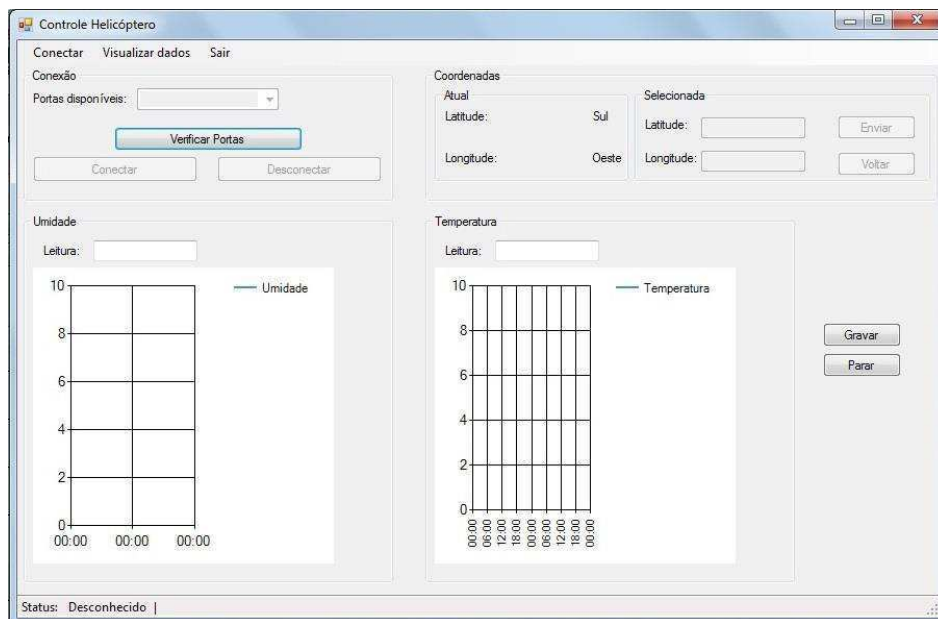


FIGURA 25: TELA PRINCIPAL DO SOFTWARE

Pode ser visto que as portas de comunicação serial são verificadas e depois selecionadas. Ao lado, as coordenadas atuais são exibidas e existe um espaço que são selecionadas as coordenadas que se deseja executar a coleta dos dados de temperatura e umidade.

São apresentados os gráficos para plotar os dados do sensor de temperatura e umidade, esses gráficos são atualizados de acordo com o recebimento de dados. Ainda existe a opção de gravação dos dados em um formato de texto (.txt) ao qual o nome do arquivo é a data e a hora que foi iniciada a medição dos dados. Por exemplo, *10-01-2014-15-52.txt*. Os dados são salvos no arquivo em duas colunas, onde a primeira é apresentada o valor da temperatura e a segunda coluna, o valor da umidade relativa do ar.

O *software* ainda possui uma funcionalidade onde podem ser vistos os dados salvos em outras medições e esses dados são plotados em um gráfico semelhante ao existente na tela principal do *software*. Esta tela de visualização dos dados é apresentada na figura 26.

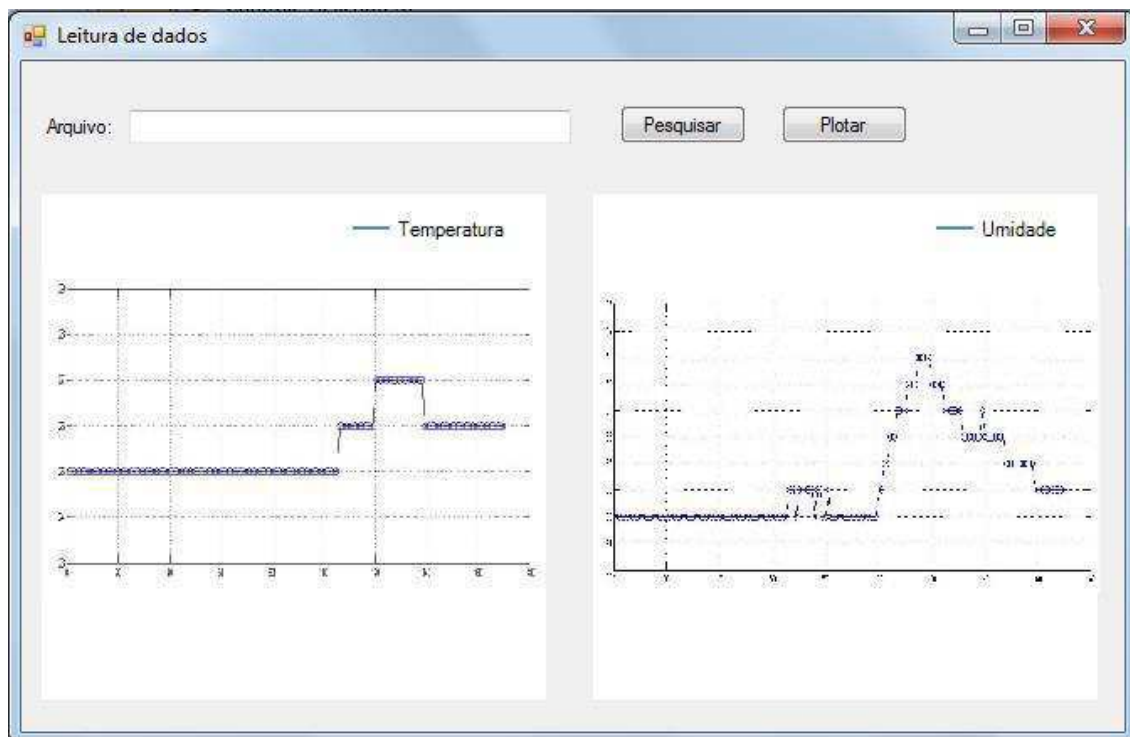


FIGURA 26: REPRESENTAÇÃO DA TELA DE VISUALIZAÇÃO DOS DADOS SALVOS

4 CONSIDERAÇÕES FINAIS

Foi percebido que é possível a implementação de um veículo aéreo não tripulado de forma simples e com baixo orçamento (R\$ 400,00), sendo possível utilizá-lo para outras aplicações, como de patrulhamento, com a adição de uma câmera.

Um dos problemas encontrado foi na idealização de um circuito simples, leve e pequeno que pudesse ser acoplado ao helicóptero ao qual o peso não prejudicasse na atividade dos motores realizando as funções de decolagem e na trajetória a ser realizada. Pois, quanto mais peso é incorporado ao helicóptero, os motores exigirão mais corrente e consequentemente as baterias descarregarão mais rápido que o previsto, tendendo a não realizar a tarefa ao qual foi submetido.

A aquisição dos dados da temperatura e umidade foi feita de uma forma mais modesta, utilizando um sensor que apresenta as duas opções de medição, a temperatura

e a umidade, ao qual as medidas não ficam tão precisas prejudicando a análise que deve ser realizada. Para um projeto a nível de produto, o ideal seria a utilização de dois sensores que exercesse as medições de forma independentes, com isso a confiabilidade do sistema seria bem maior.

Para um próximo projeto será feita a implementação em um sistema de dimensões maiores e com motores mais robustos.

REFERÊNCIAS

- Alves, S., **A Matemática do GPS**, Revista do professor de matemática 59, IME-SP, 2006
- APC220 < [http://www.dfrobot.com/wiki/index.php/APC220_Radio_Data_Module\(SKU:TEL0005\)](http://www.dfrobot.com/wiki/index.php/APC220_Radio_Data_Module(SKU:TEL0005))>, acessado em 29 de março de 2014
- Arduino Nano, < <http://arduino.cc/en/Main/arduinoBoardNano>>, acessado em 01 de abril de 2014.
- Atmel, < <http://www.atmel.com/Images/doc8161.pdf>>, acessado em 01 de abril de 2014.
- Castro, S., **Sensores de Umidade: Caracterização e Desenvolvimento de Dispositivo Eletrônico**, UNIFEI, Itajubá-MG, 2011
- Cha, S., **RCHeli: Infrastructure for PC-Controlled Micro Helicopter**, Brown University, 2008
- Curso de GPS e Cartografia Básica, **Fundamentos GPS, Sistema de Posicionamento Global**, Instituto CEUB de Pesquisa e Desenvolvimento, 2012.
- Datasheet do ADXL345, < http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf>, acessado em 02 de abril de 2014.
- Denardin, G. W., **Apostila de Microcontroladores – Versão 0.5**, UTFPR, 2008
- Fusco, D. A., **Acionamento de uma cadeira de rodas através de um acelerômetro bi-axial como inclinômetro**, UFRGS, Porto Alegre-RS, 2010
- Ko, A. W. Y., Lau, H. Y. K., **Intelligent Robot-assisted Humanitarian Search and Rescue System**, The University of Hong Kong, International Journal of Advanced Robotic Systems, Vol. 6, No. 2, pp.121-128, 2009
- Mishra, P., **Design and Fabrication of Coaxial Micro Helicopter**, Degree Bachelor, Department of Mechanical Engineering, National Institute of Technology, Rourkela, India, 2013
- Nério, A. e Rodrigues, P., **Trabalho sobre Redes sem Fio (Wireless)**, 2003
- Oki, N. e Mantovani, S. C. A., TEE I-Projeto de Robôs Móveis, **Revista Elektor**, Departamento de

Engenharia Elétrica, Faculdade de Engenharia de Ilha Solteira, 2013

Pestún, I., Halás, M., Kurčík, P., **Control of an RC Helicopter Model Through USB Interface**, Slovak University of Technology, 18th International Conference on Process Control, 2011

Rocha, F. S. e Marranghello, G. F., **Propriedades de um acelerômetro eletrônico e possibilidades de uso no ensino de mecânica**, Universidade Federal de Pelotas, Lat. Am. J. Phys. Educ. Vol. 7, 2013

Rocker, P., **Tipos de Tecnologia Wireless**, 09 de janeiro de 2009

Rodrigues, B., **RISC X CISC**, 2011

Sampaio, G. S., **Funcionalidades e aplicações de acelerômetros em dispositivos computacionais embarcados**, UniFIEO, Osasco-SP, 2012

Seisan, F. Z., **Modeling and Control Of a Co-Axial Helicopter**, University of Toronto, 2012

Shigue, C. Y., **Sensores de Temperatura**, USP, Lorena-SP, 2008

Silva, R. L., **Manual Tecnologia Wireless**, MEC/SEED/DITEC/CETE, 18 de agosto de 2008

Silva Filho, G. L., Rudiger., G. T. e Nascimento, J. P. M., **Quadricóptero**, UTFPR, Curitiba-PR, 2011

Tay, T. T., Chua, Y. L. e Iqbal, M. J., **A Proposed RC Helicopter Based Control Theory Trainer for Diploma Students**, International Conference on Teaching and Learning in Higher Education (ICTLHE 2012) in conjunction with RCEE & RHED, 2012

ANEXO A

Lista de Material

1. **Helicóptero RC Coaxial**
2. *Arduino Nano*
3. **GPS Neo-6**
4. **Acelerômetro ADXL345**
5. **Par de comunicador sem fio, APC220**
6. **Sensor de umidade e temperatura (DHT11)**

ANEXO B

Estação de base (Tela Principal)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.IO.Ports;
using System.Threading;

namespace WindowsFormsApplication1
{
    public partial class tela_inicial : Form
    {
        SerialPort porta; //Instanciando a porta serial.
        string[] portas; //Variável das portas disponíveis

        StreamWriter wr;
        string entrada;
        string temp;
        string umid;

        string latitude;
        string longitude;
        string lat_base;
        string lon_base;

        // delegate is used to write to a UI control from a non-UI thread
        private delegate void SetTextDeleg(string text);

        int X = 0;

        public tela_inicial()
        {
            InitializeComponent();
            permissoes(false, false, false, true, false, false, false, false);
            porta = new SerialPort();
            chart_temperatura.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
            chart_umidade.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        }

        //Verifica as portas que estão disponíveis e mostra nos ComboBox
        private void portas_disponiveis()
        {
            portas = SerialPort.GetPortNames();
        }
    }
}

```

```

if (portas.Length > 0)
{
    cb_portas.Items.Clear();
    cb_portas.SelectedIndex = -1;
    //cb_BR.SelectedIndex = -1;

    foreach (string s in SerialPort.GetPortNames())
    {
        cb_portas.Items.Add(s);
    }
    defineStatus(false); //Definição da condição da Barra de Status
    permissoes(true, true, false, true, false, false, false, false);
//
}

else
{
    defineStatus(false);
    permissoes(false, false, false, true, false, false, false,
false);
}
}
private void defineStatus(bool status)
{
    if (status == true)
    {
        statusLabel.Text = "Conectado";
        statusLabel.ForeColor = Color.Green;
    }
    else
    {
        statusLabel.Text = "Desconectado";
        statusLabel.ForeColor = Color.Red;
    }
}

private void permissoes(bool portas, bool conectar, bool desconectar,
    bool verifica, bool latitude, bool longitude, bool envia, bool
apaga)
{
    cb_portas.Enabled = portas;
    bt_Conectar.Enabled = conectar;
    bt_Desconectar.Enabled = desconectar;
    // cb_BR.Enabled = baudrate;
    bt_VP.Enabled = verifica;
    tB_latitude.Enabled = latitude;
    tB_longitude.Enabled = longitude;
    bt_Envia.Enabled = envia;
    bt_volta.Enabled = apaga;
}

private void bt_Desconectar_Click(object sender, EventArgs e)
{
    if (porta != null && porta.IsOpen)
    {
        porta.Close();
        porta = null;
        defineStatus(false);
        portas_disponiveis();
    }
}

```

```

    }
}

private void bt_Conectar_Click(object sender, EventArgs e)
{
    if (cb_portas.SelectedIndex != -1)
    {
        porta.PortName = cb_portas.SelectedItem.ToString();
        porta.BaudRate = 9600;
    }
    else
    {
        throw new Exception("Selecionar a Porta e o Baud Rate para
comunicação.");
    }

    if (!porta.IsOpen)
    {
        porta.Open();
        permissoes(false, false, true, false, true, true, true, true);
        defineStatus(true);
    }
}

private void bt_VP_Click(object sender, EventArgs e)
{
    portas_disponiveis();
}

e) private void tela_inicial_FormClosed(object sender, FormClosedEventArgs
{
    if (porta != null && porta.IsOpen)
    {
        porta.Close();
        porta = null;
        defineStatus(false);
        portas_disponiveis();
    }
}

private void bt_Envia_Click(object sender, EventArgs e)
{
    porta.WriteLine("A" + tB_latitude + "0" + tB_longitude);
}

private void conectarToolStripMenuItem_Click(object sender, EventArgs e)
{
    aparece(false, false, false, false, false);
}

private void aparece(bool tem, bool umi, bool grava, bool stop, bool tempo)
{
    gB_temperatura.Enabled = tem;
    gB_umidade.Enabled = umi;
    bt_grava.Enabled = grava;
    bt_para.Enabled = stop;
}

```



```

        timer.Enabled = tempo;
    }

    private void sairToolStripMenuItem_Click(object sender, EventArgs e)
    {
        bt_Desconectar_Click(sender, e);
        conectarToolStripMenuItem_Click(sender, e);
    }

    void sp_DataReceived(object sender, SerialDataReceivedEventArgs e)
    {
        Thread.Sleep(500);
        entrada = porta.ReadLine();
        this.BeginInvoke(new SetTextDeleg(si_DataReceived), new object[] {
entrada });
    }

    private void si_DataReceived(string entrada)
    {
        if (entrada.StartsWith("T"))
        {
            temp = entrada.Substring(entrada.IndexOf("T") + 1,
entrada.IndexOf("U") - 1);
            umid = entrada.Substring(entrada.IndexOf("U") + 1);
            tB_temperatura.Text = temp.Trim();
            tB_umi.Text = umid.Trim();
        }

        if (entrada.StartsWith("A"))
        {
            latitude = entrada.Substring(entrada.IndexOf("A") + 1,
entrada.IndexOf("O") - 1);
            longitude = entrada.Substring(entrada.IndexOf("O") + 1);
            label_lattitude.Text = latitude.ToString();
            label_longitude.Text = longitude.ToString();
        }
    }
}

private void timer_Tick(object sender, EventArgs e)
{
    Decimal temperatura = Convert.ToDecimal(temp);
    Decimal umidade = Convert.ToDecimal(umid);
    if (chart_temperatura.Series[0].Points.Count > 6)
    {
        chart_temperatura.Series[0].Points.RemoveAt(0);
        chart_temperatura.Update();
    }
    if (chart_umidade.Series[0].Points.Count > 6)
    {
        chart_umidade.Series[0].Points.RemoveAt(0);
        chart_umidade.Update();
    }

    chart_temperatura.Series[0].Points.AddXY(DateTime.Now, temperatura);
    chart_umidade.Series[0].Points.AddXY(DateTime.Now, umidade);
}
}

```

```
private void tela_inicial_Load(object sender, EventArgs e)
{
    porta.Handshake = Handshake.None;
    porta.DataReceived += new
SerialDataReceivedEventHandler(sp_DataReceived);
    porta.ReadTimeout = 500;
    porta.WriteTimeout = 500;
}

private void dadosToolStripMenuItem_Click(object sender, EventArgs e)
{
    tela_visualiza tv = new tela_visualiza();
    tv.Show();
}

private void bt_grava_Click(object sender, EventArgs e)
{
    //" + DateTime.Now.ToString("dd-MM-yyyy-hh-mm") + "
    wr = new StreamWriter(@"C:\Users\Caio Luiz\Desktop\Arquivo.txt",
true);
    wr.WriteLine("T" + temp + "U" + umid);
}

private void bt_para_Click(object sender, EventArgs e, StreamWriter wr)
{
    wr.Close();
}

private void bt_volta_Click(object sender, EventArgs e)
{
}

}
}
```

ANEXO C

Estação de Base (Tela de visualização)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace WindowsFormsApplication1
{
    public partial class tela_visualiza : Form
    {
        DialogResult dr;
        string arquivo;
        string dados;
        string temp;
        string umid;

        int X = 0;

        public tela_visualiza()
        {
            InitializeComponent();
            chart_leitura_temp.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
            chart_leitura_umidade.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;
        }

        private void bt_pesquisa_Click(object sender, EventArgs e)
        {
            dr = this.openFileDialog.ShowDialog();

            if (dr == System.Windows.Forms.DialogResult.OK)
            {
                arquivo = openFileDialog.FileName;
                tB_arquivo.Text = arquivo;
            }
        }

        private void bt_plot_Click(object sender, EventArgs e)
        {
            StreamReader str = new StreamReader(arquivo);

            while ((dados = str.ReadLine()) != null)
            {
                temp = dados.Substring(dados.IndexOf("T") + 1, dados.IndexOf("U")
- 1);

```

```
        umid = dados.Substring(dados.IndexOf("U") + 1);

        Decimal temperatura = Convert.ToDecimal(temp);
        Double umidade = Convert.ToDouble(umid);

        chart_leitura_temp.Series[0].Points.AddXY(X, temperatura);
        chart_leitura_umidade.Series[0].Points.AddXY(X, umidade);
        chart_leitura_umidade.Update();
        chart_leitura_temp.Update();
    }
}
}
```

ANEXO D

Arduino

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include <ADXL345.h>
#include <String.h>

SoftwareSerial gps(10,11);
ADXL345 accel;

int rotorP_1 = 1; //Rotor Principal superior
int rotorP_2 = 2; //Rotor Principal Inferior
int calda = 3;   //Rotoro de Calda

int i=0;
int tamanho = 0;
int statusPin = 2;

char la[9]; //Latitude
char hla; //Direção
```

```
char lo[9]; //Longitude
char hlo; //Direção
char la_dest[9]; //Latitude de destino
char lo_dest[9]; //Longitude de destino

void setup()
{
  Serial.begin(9600); //Comunicação com o APC220
  gps.begin(9600); //GPS
  Wire.begin();
  pinMode(statusPin, OUTPUT);
  pinMode(rotorP_1, OUTPUT);
  pinMode(rotorP_2, OUTPUT);
  pinMode(calda, OUTPUT);

  //Iniciando os motores
  analogWrite(rotorP_1, 0);
  analogWrite(rotorP_2, 0);
  analogWrite(calda, 0);

  //Iniciando o acelerômetro
  acel.SetRange(16, true);
  acel.EnableMeasurements();
}

void loop()
{
  char entrada[100] = 0; //Entrada do GPS

  while(Serial.available()){
    entrada[i]=Serial.read();
    i++;
  }
```

```
tamanho = length(entrada);
for(i==0;i<=tamanho;i++)
{
  if(entrada[0]=="$")
  {
    if(entrada[5]=="A")
    {
      for(int j==0;j<=8;j++)
      {
        la[j]=entrada[17+j];
        Serial.write(la[j]);
      }
      hla = entrada[27];
      for(int j==0;j<=8;j++)
      {
        lo[j]=entrada[29+j];
        Serial.write(lo[j]);
      }
      Serial.println();
      hlo=entrada[39];
    }
  }
  else if(entrada[0]=="A")
  {
    for(int j ==0;j<=8;j++)
    {
      la_dest[j] = entrada[1+j];
    }
    for(int j == 0;j<=8;j++)
    {
      lo_dest[j] = entrada[11+j];
    }
  }
}
```

```
void trajeto();  
}  
  
void trajeto()  
{  
    if(la!=la_dest)  
    {  
  
    }  
}  
}
```