



# **ANÁLISE DE REQUISITOS DE HARDWARE EM PROJETO DE UAV QUADROTOR**

**Trabalho de Conclusão de Curso**

**Engenharia de Computação**

**Aluno: David Alain do Nascimento**  
**Orientador: Prof. Dr. Sergio Campello Oliveira**



**UNIVERSIDADE  
DE PERNAMBUCO**

---

**Universidade de Pernambuco  
Escola Politécnica de Pernambuco  
Graduação em Engenharia de Computação**

**DAVID ALAIN DO NASCIMENTO**

**ANÁLISE DE REQUISITOS DE  
*HARDWARE* EM PROJETO DE UAV  
QUADROTOR**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

**Recife, outubro de 2011.**

## MONOGRAFIA DE FINAL DE CURSO

### Avaliação Final (para o presidente da banca)\*

No dia 23 de Dezembro de 2011, às 9:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente DAVID ALAIN DO NASCIMENTO, orientado pelo professor Sérgio Campello Oliveira, sob título ANÁLISE DE REQUISITOS DE HARDWARE EM PROJETO DE UAV QUADROTOR, a banca composta pelos professores:

**Marcílio André Felix Feitosa**

**Sérgio Campello Oliveira**

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada       Aprovada com Restrições\*       Reprovada

e foi-lhe atribuída nota: 10,0 ( dez )

\*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 3 dias para entrega da versão final da monografia a contar da data deste documento.

MARCÍLIO ANDRÉ FELIX FEITOSA

SÉRGIO CAMPELLO OLIVEIRA

---

*Dedico este trabalho a minha mãe, Avanilda e a minha avó, Severina, que sempre acreditaram no meu potencial e que são o fator mais importante de eu ser a pessoa que sou hoje.*

---

# Agradecimentos

Agradeço primeiramente a Deus pela minha saúde, inteligência, persistência e esperança para conseguir meus objetivos.

Agradeço a minha mãe que, sempre, desde criança, me incentivou a estudar e me ensinou que estudar nos dá um futuro melhor, e a minha avó que é minha segunda mãe.

Agradeço a minha ex-vizinha e amiga, Maria das Graças, que me ajudou nos momentos financeiramente difíceis com o transporte diário para a universidade.

Agradeço ao orientador Sergio Campello Oliveira por me guiar neste processo de aprendizagem acadêmica e profissional.

Aos colegas Gildo Ferrúcio e Marcos Gabriel do grupo de pesquisa *Swarm Robots*, que me ajudaram com conhecimentos adquiridos em suas pesquisas.

Ao colega Cristóvão Rufino pelas sugestões de melhorias na escrita desta monografia.

Aos colegas Fábio Leite e Helmut Kemper que disponibilizaram placas Arduino Mega para realização dos experimentos.

Agradeço aos meus amigos Bruno Lins, Caio Bernardes, Denis Martins, Diego Marconi, Erick Barboza, Marcelo Lacerda, Paulo Roger, Rafael Farias, Ricardo Teixeira e Rodrigo Carneiro, que, com suas habilidades, me mostraram como é ser bom no que se faz.

---

# Resumo

O uso de UAVs têm se difundido pelo mundo pela possibilidade de uso em aplicações diversas e pelo relativo baixo custo. Avanços tecnológicos e miniaturização permitem embarcar vários dispositivos em pequenos UAVs, aumentando suas funcionalidades. Para que os UAVs possam realizar suas tarefas de forma efetiva, eles precisam estar dotados de algumas funcionalidades como controle de seus motores, percepção de invasores, percepção de obstáculos, geolocalização, monitoramento de saúde e capacidade de comunicação com outros UAVs. Este trabalho tem como objetivo analisar os dispositivos necessários a construção de UAV quadrotor para implementação de um enxame de UAVs com foco em aplicações de monitoramento territorial.

---

# Abstract

Use of unmanned aerial vehicles (UAV) has been spreading around the world for possible use in many applications and their related low cost. Technological advances and miniaturization allow embedding several devices in small UAVs, increasing their features. Wherefore the UAVs can do actually their tasks, they need have many skills as motor control, perception of invaders, perceptions of obstacles, geolocation, health monitoring and ability to communicate with other UAVs. This work will analyze devices that are needed to build UAVs quadrotor to a swarm UAV implementation for territorial monitoring applications.

---

# Sumário

## Conteúdo

<b>Capítulo 1 Introdução</b>	<b>1</b>
<b>Capítulo 2 UAV quadrotor</b>	<b>4</b>
2.1 UAV	4
2.2 UAV Quadrotor	6
2.3 Enxame de UAVs	8
<b>Capítulo 3</b>	<b>9</b>
<b>Sistemas Embarcados</b>	<b>9</b>
3.1 Microcontrolador	10
<b>Capítulo 4</b>	<b>12</b>
<b>Dispositivos utilizados em UAV quadrotor</b>	<b>12</b>
4.1 Microcontrolador	14
4.2 Magnetômetro	18
4.3 Acelerômetro	19
4.4 Giroscópio	20
4.5 Altímetro	22
4.6 GPS	23
4.7 Motores	24
4.7.1 <i>Brushed</i>	25
4.7.2 <i>Brushless</i>	26
4.8 ESC	27
4.9 Bateria	30
4.10 Comunicação sem fio de longo alcance	32

---

<b>Capítulo 5</b>	<b>34</b>
<b>Experimento controlado para validação do desempenho da plataforma Arduino Mega</b>	<b>34</b>
5.1 Microcontrolador Atmel ATmega1280	35
5.2 <i>Particle Swarm Optimization</i>	35
5.3 Objetivos	36
5.3.1 Objetivo global	36
5.3.2 Objetivos do estudo	36
5.3.3 Questão:	36
5.4 Planejamento	36
5.4.1 Definição das hipóteses	36
5.4.2 Descrição da instrumentação	37
5.4.3 Seleção do contexto	37
5.4.4 Variáveis	38
5.4.5 Análise quantitativa	39
5.4.6 Tipo de projeto	39
5.4.7 Coleta, validação e armazenamento de dados	39
5.4.8 Análise dos dados	39
5.5 Execução	40
5.6 Resultados obtidos	41
5.6.1 Validação das hipóteses	44
<b>Capítulo 6</b>	<b>45</b>
<b>Resultados</b>	<b>45</b>
6.1 Dispositivos propostos	45
6.1.1 Microcontrolador	45
6.1.2 Magnetômetro	45

---

6.1.3	Acelerômetro	45
6.1.4	Giroscópio	46
6.1.5	Altímetro	46
6.1.6	GPS	46
6.1.7	Motores	46
6.1.8	ESCs	46
6.1.9	Bateria	46
6.1.10	Comunicação sem fio de longo alcance	46
<b>Capítulo 7 Conclusão e Trabalhos Futuros</b>		<b>47</b>
7.1	Trabalhos futuros	48
<b>Bibliografia</b>		<b>50</b>
<b>Apêndice A</b>		<b>55</b>
<b>Implementação em C++ do algoritmo PSO</b>		<b>55</b>

---

# Índice de Tabelas

*Tabela 1 - Comparativo entre os microcontroladores ATmega1280, ATmega2560, LPC1343 e LPC1768.*

\_\_\_\_\_ 17

*Tabela 2 - Relação dos tipos de bateria - capacidade, volume e massa.* \_\_\_\_\_ 30

*Tabela 3 - Resultados das simulações sem a utilização do PSO.* \_\_\_\_\_ 41

*Tabela 4 - Resultados das simulações com a utilização do PSO.* \_\_\_\_\_ 43

---

# Índice de Figuras

<i>Figura 1 – MQ-1 Predator - UAV da Força Aérea Americana.....</i>	<i>5</i>
<i>Figura 2 – COLIBRI - Projeto de UAV desenvolvido pelo Computer Vision Group - Universidad Politécnica de Madrid.....</i>	<i>6</i>
<i>Figura 3 - Modo de voo com configuração em X (a) e com configuração em + (b). ....</i>	<i>7</i>
<i>Figura 4 - Esquema de comandos para controle de um quadrotor.....</i>	<i>7</i>
<i>Figura 5 - Diagrama esquemático dos dispositivos utilizados em quadrotor, extraído de (Melo, 2010). .....</i>	<i>13</i>
<i>Figura 6 - Representação 3D dos dispositivos utilizados no projeto (ArduCopter, 2011). ....</i>	<i>14</i>
<i>Figura 7 - Plataforma Arduino Mega 2560. ....</i>	<i>15</i>
<i>Figura 8 - Plataforma MBED. ....</i>	<i>15</i>
<i>Figura 9 - Plataforma NXP LPCXpresso LPC1343.....</i>	<i>16</i>
<i>Figura 10 - Placa de sensoriamento com magnetômetro HMC5883L utilizada o projeto AeroQuad... </i>	<i>19</i>
<i>Figura 11 - Placa com acelerômetro BMA180 utilizado no projeto AeroQuad.....</i>	<i>20</i>
<i>Figura 12 - Princípio do giroscópio.....</i>	<i>21</i>
<i>Figura 13 - Placa com giroscópio ITG-3200 utilizado no projeto AeroQuad. ....</i>	<i>21</i>
<i>Figura 14 - Placa com o sensor de pressão Bosch BMP085 utilizado no projeto AeroQuad.....</i>	<i>23</i>
<i>Figura 15 - Exemplo obtenção posicionamento baseado em triangulação. ....</i>	<i>24</i>
<i>Figura 16 - Funcionamento de um motor de dois polos com escovas.....</i>	<i>25</i>
<i>Figura 17 - Funcionamento de um motor de dois polos sem escovas.....</i>	<i>26</i>
<i>Figura 18 - Modelo de ESC com entrada de controle por PWM e três fases de saída para alimentação do motor BLDC. ....</i>	<i>28</i>
<i>Figura 19 - Forma de onda gerada para alimentação de cada fase.....</i>	<i>28</i>
<i>Figura 20 - Topologias de rede do padrão IEEE 802.15.4.....</i>	<i>33</i>
<i>Figura 21 - Tempos de execução das amostras de simulações sem a utilização do PSO.....</i>	<i>42</i>
<i>Figura 22- Tempos de execução das amostras de simulações com a utilização do PSO. ....</i>	<i>43</i>

---

# Tabela de Símbolos e Siglas

AC - *Alternating Current*

ADC – *Analogic-to-Digital Converter*

AES - *Advanced Encryption Standard*

API - *Application Programming Interface*

BLDC - *Brushless Direct Current*

CISC - *Complex Instruction Set Computer*

DAC – *Digital-to-Analogic Converter*

DC - *Direct Current*

EEPROM - *Electrically-Erasable Programmable Read-Only Memory*

ESC - *Eletronic Speed Controller*

FDD - *Full-Function Device*

FPGA - *Field Programmable Gate Array*

GPS - *Global Positioning System*

GPIO - *General Purpose Input/Output*

HID – *Human Interface Device*

IDE - *Integrated Development Environment*

IEEE - *Institute of Electrical and Electronics Engineers*

IMU - *Inertial Measurement Unit*

I2C - *Inter-Integrated Circuit*

ISM - *Industrial, Scientific and Medical*

LED - *Light Emitting Diode*

LiPo - *Lithium-ion polymer*

MEMS - *Microelectromechanical systems*

---

MOSFET - *Metal Oxide Semiconductor Field Effect Transistor*

PID – *Proporcional - Integral – Derivativo*

PSO – *Particle Swarm Optimization*

PWM - *Pulse-Width Modulation*

RAM - *Random Access Memory*

RDD - *Reduced-Function Devices*

RISC - *Reduced Instruction Set Computer*

ROM - *Read Only Memory*

SPI - *Serial Peripheral Interface*

UAV - *Unmanned Aerial Vehicle*

VTOL - *Vertical Take-Off and Landing*

WLAN - *Wireless Local Area Network*

WPAN - *Wireless Personal Area Networks*

---

# Capítulo 1

## Introdução

Em situações, como por exemplo, em missões militares, onde a vigilância por aeronave tripulada é inviável de ser realizada por conta de riscos, custos e outros fatores, faz-se necessário o uso de veículos aéreos não tripulados para realizar essa vigilância de forma economicamente eficiente. O uso de veículos aéreos não tripulados - UAV (do inglês, *Unmanned Aerial Vehicle*), tem se difundido pelo mundo pela possibilidade de uso em aplicações diversas e pelo relativo baixo custo. Avanços tecnológicos e miniaturização permitem embarcar vários dispositivos em pequenos UAVs, aumentando suas funcionalidades (Strojny, 2009).

Para que os UAVs possam realizar suas tarefas de forma efetiva, eles precisam estar dotados de algumas funcionalidades como controle de seus motores, percepção de invasores, percepção de obstáculos, localização geográfica, monitoramento de saúde e capacidade de comunicação.

Alguns projetos de quadricópteros UAV (AeroQuad, 2011), (ArduCopter, 2011), (Multipilot-8, 2011) e (Multipilot-32, 2011) utilizam dispositivos sensores como magnetômetro, acelerômetro, giroscópio, barômetro e GPS, e outros como motores, controlador eletrônico de velocidades para os motores, câmera, comunicação sem fio de longo alcance, bateria e microcontrolador. Este projeto pretende investigar se apenas estes dispositivos são necessários ou se outros dispositivos são necessários para cumprir com os requisitos descritos.

Em (AeroQuad, 2011), (Multipilot-8, 2011) e (ArduCopter, 2011) o processamento das informações de gerenciamento e controle são feitas através da plataforma Arduino Mega 2560 (Arduino, 2011) que possui como unidade de processamento o microcontrolador de arquitetura Atmel megaAVR (Atmel, 2011). Em (Multipilot-32, 2011) é utilizada uma plataforma própria que possui como unidade de processamento um microcontrolador de arquitetura ARM. Analisando-se as duas arquiteturas, percebe-se que a simplicidade da plataforma Arduino é uma grande vantagem para sua escolha em projetos de sistemas embarcados. Porém, levando-

---

se em conta o desempenho e outros fatores importantes como recursos nativos disponíveis, percebe-se que um microcontrolador de arquitetura ARM possui maior vantagem.

Este trabalho estima o uso de um microcontrolador ARM para realização do controle de UAV quadrotor em substituição a plataforma Arduino Mega utilizada em outros projetos, para tanto serão feitas algumas análises sobre estas duas arquiteturas mostrando algumas vantagens e desvantagens de cada uma.

A escolha do microcontrolador para realização do processamento das informações de um UAV quadrotor é uma importante decisão no projeto. Para tanto é necessário saber se o microcontrolador a ser utilizado é capaz de realizar todas as atividades sem que se tenham atrasos nas respostas.

Alguns dos projetos de UAV quadrotor analisados neste trabalho utilizam a plataforma Arduino Mega como unidade de processamento, porém para implementação de um enxame de UAVs quadrotor é necessário que a unidade de processamento tenha capacidade de processar dados de rota dinâmica de voo e coordenação com os outros UAVs, para tanto foi realizado um experimento controlado com a plataforma Arduino Mega para validação da sua capacidade de processamento para este fim.

O projeto de um UAV quadrotor envolve a escolha de sensores e atuadores, suas partes mecânicas, processamento de sinais e algoritmo de estabilização de comportamento durante o voo.

Este trabalho é um estudo preliminar sobre os dispositivos necessários em projetos de UAV quadrotor. Para isso serão estudados os requisitos necessários em um projeto de UAV que cumpra com todas as funcionalidades descritas anteriormente. Algumas partes não serão estudadas como as hélices, já que são recomendadas de acordo com os motores, e também o projeto da estrutura mecânica onde serão montados os dispositivos, conhecida como *frame*.

Este documento está dividido em seis capítulos, que estão organizados da seguinte forma: no Capítulo 2 serão abordados aspectos sobre UAV quadrotor. Em seguida, No Capítulo 3 serão descritos sistemas embarcados e suas aplicações. O

---

Capítulo 4 é um estudo sobre os principais dispositivos utilizados em projetos de UAV quadrotor. No Capítulo 5 é mostrado um experimento controlado para validação da capacidade de processamento da plataforma Arduino Mega para aplicação do controle de UAV quadrotor. E por fim, o Capítulo 6 mostra as conclusões obtidas acerca deste trabalho e propostas de trabalhos futuros.

---

# Capítulo 2

## UAV quadrotor

O uso de UAVs (do inglês – *Unmanned Aerial Vehicle*) têm se difundido pelo mundo pela possibilidade de uso em aplicações diversas e pelo relativo baixo custo. Avanços tecnológicos relacionados a baterias, sensores e processamento de sinais permitem a construção de UAVs pequenos, leves e com custos acessíveis. Dessa forma vários pesquisadores exploram novas oportunidades e aplicações para UAVs e resolvem questões desafiadoras sobre projeto, controle e navegação autônoma (Hoffmann, 2010) (Strojny, 2009).

### 2.1 UAV

UAV é uma aeronave não tripulada, controlada remotamente ou autocontrolada, que possui características que a torna uma boa solução em relação ao uso de aeronave tripulada.

Idealizado inicialmente para fins militares, os primeiros UAVs foram criados para serem usados em missões que eram perigosas para humanos, em que a aeronave carregava bombas para serem lançadas em locais específicos, sem colocar em risco a vida de um piloto. Porém atualmente há outros que foram construídos para uso em inteligência militar, apoio aéreo e controle de tiro de artilharia, como também para outras atividades como patrulhamento urbano, ambiental e de fronteiras, atividades de busca e resgate, entre outros (Wikipedia, Veículo\_aéreo\_não\_tripulado, 2011).

UAVs armados de uso militar, por exemplo, o mostrado na Figura 1, também chamados de UAV de combate, são vantajosos em relação às aeronaves tripuladas, pelo fato de não haver risco de constrangimento diplomático caso a aeronave caia e o piloto seja capturado, como também a morte do piloto. UAVs deste tipo possuem capacidade de multiplicação de forças, bombardeio estratégico, melhor inteligência,

---

busca e reconhecimento, o que os torna uma melhor opção para uso em ataques (Sullivan, 2005).



**Figura 1 – MQ-1 Predator - UAV da Força Aérea Americana.**

Atualmente UAVs de uso comercial, estão em ascensão, com aplicações em vigilância aérea; exploração e produção de óleo, gás e minerais; transporte, entre outros (Wikipedia, Unmanned\_aerial\_vehicle, 2011).

O projeto COLIBRI, mostrado na Figura 2, (Mejias, Correa, Mondragon, & Campoy, 2007) é um exemplo de projeto de UAV desenvolvido para vigilância aérea, que usa algoritmos de visão computacional para vigilância de linhas de transmissão e isoladores, como também recursos visuais 2D para controle dos deslocamentos vertical, lateral e longitudinal de um helicóptero autônomo. O COLIBRI possui navegação baseada em visão estérea, o que lhe dá algumas capacidades interessantes, como estimativa de altitude.



Figura 2 – COLIBRI - Projeto de UAV desenvolvido pelo *Computer Vision Group* - Universidad Politécnica de Madrid.

## 2.2 UAV Quadrotor

Também chamado de quadricóptero, o UAV quadrotor é um tipo de aeronave multi-rotor não-coaxial em distribuição borboleta que consegue fazer decolagem e pouso verticais (VTOL, do inglês - *Vertical Take-Off and Landing*). O controle de atitude de voo do quadrotor pode ser realizado através do ajuste da velocidade dos quatro rotores. A força de levantamento de quatro rotores é mais uniforme do que de um único rotor, e assim, a atitude de voo é mais estável. Comparado com aeronaves de asa fixa, os requisitos de decolagem do quadrotor são menores, têm uma melhor adaptação ambiental, entre outros. Dessa forma é fácil perceber que há diversas vantagens no uso de quadrotor em relação a outros tipos de aeronaves (Li & Li, 2011 ).

Um quadrotor possui um esqueleto que pode ser utilizado com configuração em X ou configuração em +, relativamente a direção preferencial de voo, como pode ser visto na Figura 3. Nas suas extremidades ficam os motores e no meio, por conta da estabilidade, ficam os dispositivos de controle, sensores, bateria, etc. Esse esqueleto é construído com materiais leves e resistentes, como alumínio ou fibra de carbono.

Como ilustrado na Figura 3, cada motor tem uma rotação em sentido específico, de forma que um motor sempre gira no sentido contrário do seu

adjacente. Isso deve ser feito para que se mantenha a estabilidade durante o voo, visto que se todos girassem no mesmo sentido seria criado um torque, assim o quadrotor acabaria girando em sentido contrário e perderia a estabilidade.

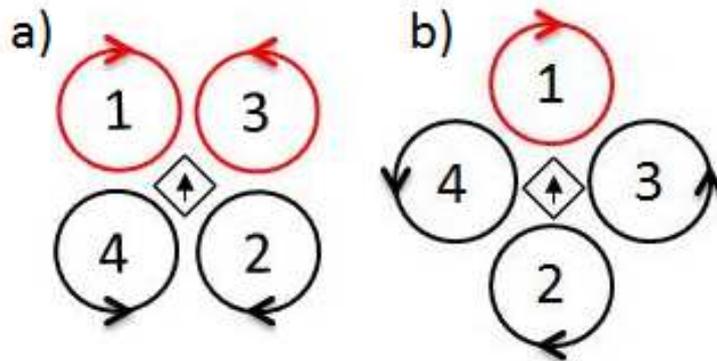


Figura 3 - Modo de voo com configuração em X (a) e com configuração em + (b).

Como já dito anteriormente, o controle de um quadrotor é feito através da escolha da velocidade dos seus motores, e os seus movimentos são realizados através de alguns comandos: *Throttle* (aceleração) que é o movimento vertical, *Yaw* (guinada) é o movimento de rotação no eixo Z, *Roll* (rolagem) é o movimento de rotação no eixo X, e *Pitch* (arfagem) é o movimento de rotação no eixo Y. Estes comandos podem ser visualizados na Figura 4. Dessa forma o controlador deve ajustar a velocidade do motor para que se consigam os valores desejados dos *Throttle*, *Yaw*, *Roll* e *Pitch*.

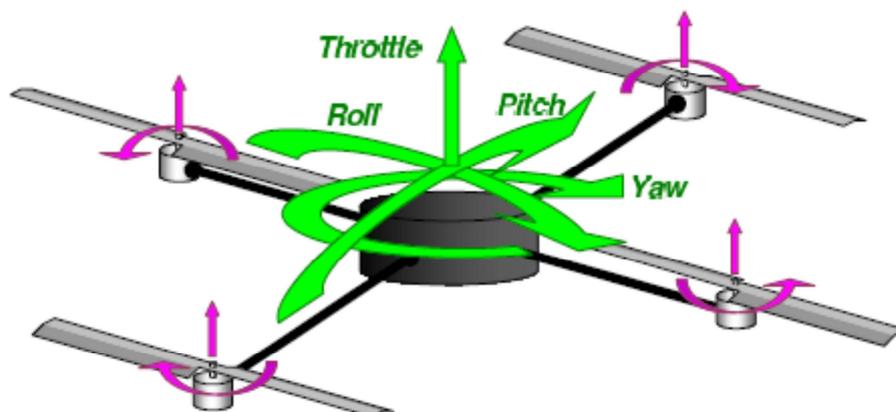


Figura 4 - Esquema de comandos para controle de um quadrotor.

---

## 2.3 Enxame de UAVs

Um novo paradigma sendo muito pesquisado atualmente é o de enxame de robôs em que são utilizados conceitos do comportamento social de insetos e outros animais para, através de simples ações, conseguir realizar tarefas complexas, de modo que estas simples unidades, individualmente, conseguem se interagir para criar sistemas inteligentes de forma coletiva.

Estudos realizados mostraram que ali existem mecanismos de coordenação não centralizada através de operações sincronizadas, no que diz respeito ao comportamento social dos insetos, e ainda, o seu funcionamento ao nível de um sistema é robusto, flexível e escalável (Camazine, et al., 2001).

Em (Sahin, 2005) são propostos os seguintes critérios para distinguir uma arquitetura de um enxame de robôs:

- indivíduos autônomos, no que diz respeito à tomada de decisão;
- grande quantidade de indivíduos, dependendo da aplicação;
- grupo de indivíduos pouco homogêneos, onde as diferenças de características tornam o sistema mais versátil e completo, utilizando menos recursos;
- indivíduos simples, ou seja, relativamente incapazes ou ineficientes para realizar suas tarefas sozinhos;
- indivíduos com sensoriamento local e capacidade de comunicação.

Enxame de UAVs é uma subárea dessa área de enxame de robôs, que através dos mesmos conceitos já descritos, são utilizados para resolução de problemas de diversos tipos, principalmente onde há dificuldade de locomoção em solo.

---

# Capítulo 3

## Sistemas Embarcados

Conceitualmente, sistemas embarcados podem ser definidos como sistemas que possuem todos os principais componentes de um pequeno computador como processador, memórias de processamento e armazenamento e interfaces de entrada e saída, e são utilizados em aplicações de propósitos específicos que realizam continuamente uma única atividade (Morimoto, 2007). Um sistema embarcado pode ser utilizado para diversas aplicações e construído baseado em diferentes tipos de unidades de processamento como microcontroladores (Souza, 2000), microprocessadores (Tanenbaum, 1999) e FPGA (Costa, 2009).

Dispositivos como Nintendo Game Boy (Fayzullin, 2011) e OIWTECH Access Point (OIWTECH, 2010) são exemplos de sistemas embarcados baseados em microprocessador. Dispositivos como *Conmutador WLAN OmniAccess 4302* (Alcatel-Lucent, 2009) é um exemplo sistema embarcado baseado em FPGA. Atualmente existem projetos que possuem um FPGA embarcado como um coprocessador para acelerar o processamento do sistema através do paralelismo do FPGA, como o (Alcatel-Lucent, 2009).

A grande maioria dos equipamentos que possuímos em nossa residência possui um microcontrolador para controlá-lo. Por exemplo, teclado do computador, monitor, disco rígido, relógio de pulso, rádio relógio, máquinas de lavar, forno de micro-ondas, telefone, etc. (Wikipedia, Microcontroller, 2011). E isso vem acontecendo devido à simplicidade de desenvolvimento de projetos com microcontroladores, através de IDEs e APIs que as empresas, que na grande maioria das vezes são os próprios fabricantes dos microcontroladores, estão desenvolvendo para facilitar a criação de novos projetos.

Na seção seguinte serão descritos conceitos sobre microcontrolador e sistemas embarcados baseados em microcontrolador.

---

## 3.1 Microcontrolador

Um microcontrolador pode ser definido como um pequeno *chip* programável, com dispositivos equivalentes ao de um computador como núcleo de processamento, memória RAM e ROM, *timers*, contadores, portas de entrada de entrada e/ou saída paralela, conversores ADC (do inglês - *Analog-to-digital converter*) e DAC (do inglês - *Digital-to-analog converter*), comunicação serial, entre outros, utilizado no controle de diversos periféricos como LEDs, botões, relés, sensores, atuadores, etc. Também considerado um computador em um chip, um microcontrolador é nomeado um dispositivo programável por conta de que toda sua lógica de operação é feita através de um programa que fica armazenado na sua memória interna (Souza, 2000) (SCHUNK & LUPPI, 2001).

Os atuais microcontroladores são baseados em duas arquiteturas: Harvard ou Von-Neumann (Souza, 2000). A principal diferença entre as duas é que a Harvard possui o barramento de dados separado do barramento de instruções, o que torna a arquitetura Harvard mais rápida, pois enquanto uma instrução é executada a próxima instrução pode ser buscada da memória, podendo inclusive os barramentos possuir larguras distintas, o que não acontece na Von-Neumann, por ser o mesmo (Souza, 2000).

Além das arquiteturas estruturais por conta dos barramentos, existem duas outras arquiteturas o qual um microcontrolador pode ter, baseado na arquitetura do núcleo de processamento, que são as arquiteturas RISC e CISC.

A arquitetura RISC (do inglês - *Reduced Instruction Set Computer*), é uma arquitetura que possui um conjunto com poucas instruções e executa uma instrução por ciclo de máquina. Outra característica é executar operações apenas entre registradores, e o acesso à memória é feito apenas para ler e gravar dados. Possui instruções de tamanho fixo igual ao tamanho da palavra de dado, o que torna a instrução mais simples de ser interpretada pelo núcleo de processamento e, por conseguinte mais rápida de ser executada (Stallings, 2003).

A arquitetura CISC (do inglês - *Complex Instruction Set Computer*), é uma arquitetura que possui uma vasta gama de instruções para execução das suas

---

operações. Esta arquitetura possui instruções que representam combinações de instruções simples, e que podem executar tarefas com um elevado grau de complexidade. Outra característica desta arquitetura é possuir instruções de tamanhos variados, por isso possui em seu núcleo de processamento um microcódigo para interpretação das instruções (Stallings, 2003).

Independentes da arquitetura a qual é baseada, os microcontroladores possuem muitos recursos comuns. A principal característica de um microcontrolador é que este possui uma memória especial chamada memória de programa, onde é armazenado o código a ser executado pelo mesmo. Por ser um dispositivo pequeno, o microcontrolador torna-se uma boa solução para projetos de sistemas embarcados que possuam recursos limitados de energia ou não necessitem de muita capacidade de processamento. Por conta disso, possuem pouca memória, normalmente algumas unidades de KB de memória RAM e centenas de KB de memória ROM ou *Flash*. Há alguns modelos que possuem muito mais memória, chegando a alguns MB de memória RAM, ROM ou Flash.

Outra característica interessante deste dispositivo são os *timers*, que podem ser usados como contadores ou temporizadores, com a vantagem de não consumir tempo de processamento do núcleo, por ser uma funcionalidade nativa implementada em *hardware*.

Por ser um dispositivo comumente utilizado para controle, ele necessita de interação com o dispositivo/equipamento que será controlado, que é feito através de suas portas de E/S. Essas portas, normalmente apelidadas por letras, são conjuntos de pinos de que possuem funções de E/S, podendo ser apenas entrada, apenas saída, bidirecional, ou com funções mais complexas como PWM e porta serial.

---

## Capítulo 4

# Dispositivos utilizados em UAV quadrotor

Neste capítulo serão descritos os principais componentes utilizados em projetos de UAV quadrotor. Estes dispositivos possuem funcionalidades importantes que provêm os dados necessários ao UAV para que este tenha percepção do ambiente, do seu estado e dos estados dos outros UAVs do enxame; tenha comunicação efetiva com os outros UAVs; e controle eficiente de voo.

A Figura 5, obtida em (Melo, 2010), mostra um diagrama esquemático dos dispositivos necessários à construção de um quadrotor, onde pode ser visto que são necessários: quatro motores BLDC (de Corrente Contínua Sem Escovas, do inglês – *Brushless Direct Current*); quatro ESCs (Controladores Eletrônicos de Velocidade, do inglês – *Electronic Speed Controllers*), sendo um para cada motor; uma unidade de processamento, que neste caso é uma placa com microcontrolador; uma unidade de sensoriamento, também chamada de IMU (Unidade de Medida Inercial, do inglês - *Inertial Measurement Unit*), que possui acelerômetro e giroscópio; módulo de comunicação sem fio; e, uma bateria do tipo LiPo 3S de 11,1V. O rádio transmissor não será utilizado, pois este trabalho é parte de uma pesquisa que visa implementar um enxame de UAVs autocontrolados.

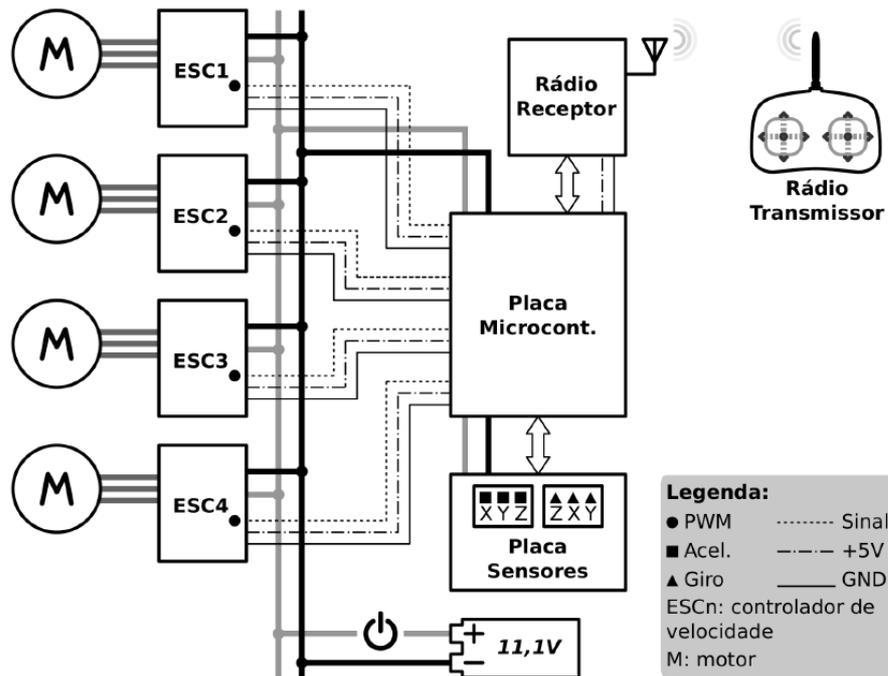


Figura 5 - Diagrama esquemático dos dispositivos utilizados em quadrotor, extraído de (Melo, 2010).

A Figura 6 mostra um exemplo tridimensional dos dispositivos utilizados no projeto (ArduCopter, 2011), onde podem ser vistos dispositivos como GPS, magnetômetro, módulo de comunicação sem fio, placa de sensores, placa Arduino, ESCs e bateria LiPo.

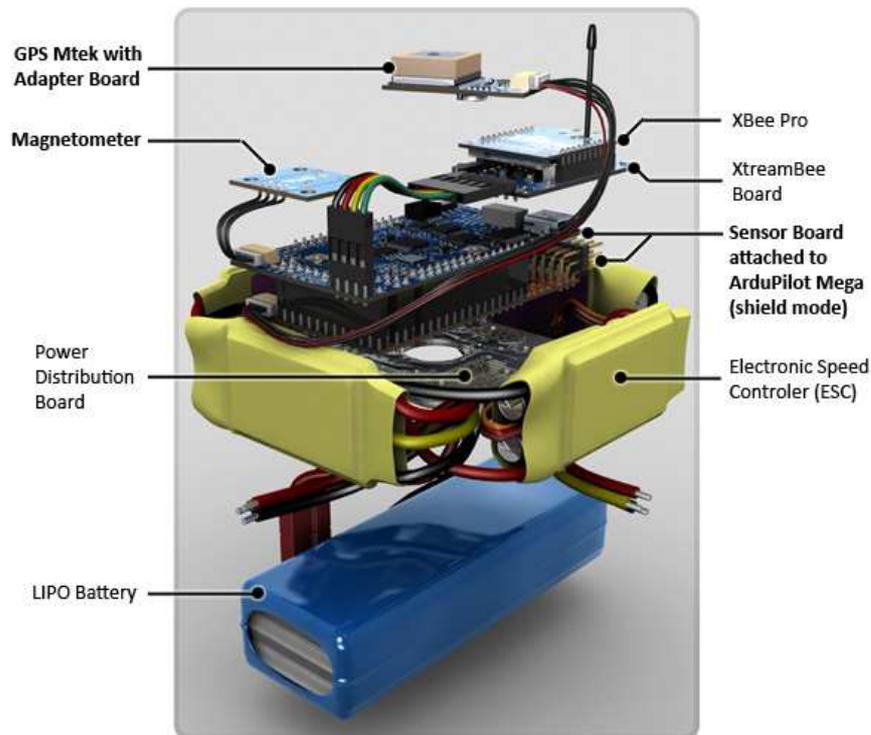


Figura 6 - Representação 3D dos dispositivos utilizados no projeto (ArduCopter, 2011).

## 4.1 Microcontrolador

Em alguns projetos de quadricópteros UAV foram utilizados microcontroladores de arquitetura Atmel MegaAVR (Atmel, 2011) e ARM Cortex-M3 (ARM, 2011). Os projetos baseados na arquitetura Atmel MegaAVR foram desenvolvidos através da plataforma de desenvolvimento Arduino Mega 2560 (Arduino, 2011), mostrada na Figura 7, que possuem diversos algoritmos implementados para auxiliar no desenvolvimento dos projetos. Dessa forma o projetista pode preocupar-se apenas com as funcionalidades do seu projeto sem precisar desenvolver algoritmos básicos para controle dos recursos disponíveis no microcontrolador. E também, caso necessite, pode fazer seus próprios algoritmos de controle dos recursos, o que torna esta plataforma uma ótima opção para construção de protótipos, tanto para os iniciantes como para os experientes.

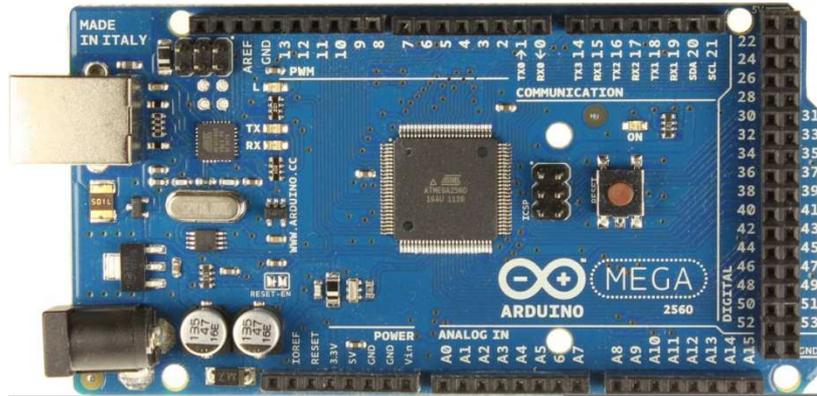


Figura 7 - Plataforma Arduino Mega 2560.

Baseados na arquitetura ARM, existem outras plataformas de desenvolvimento como a MBED (MBED, 2011), mostrada na Figura 8, e LPCXpresso (NXP-LPCXpresso, 2011), mostrada na Figura 9, que também possuem diversos algoritmos para auxiliar nos projetos.

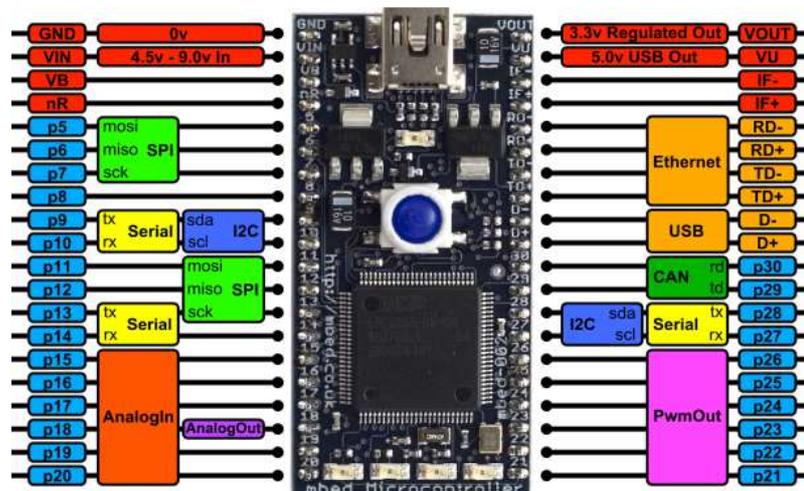
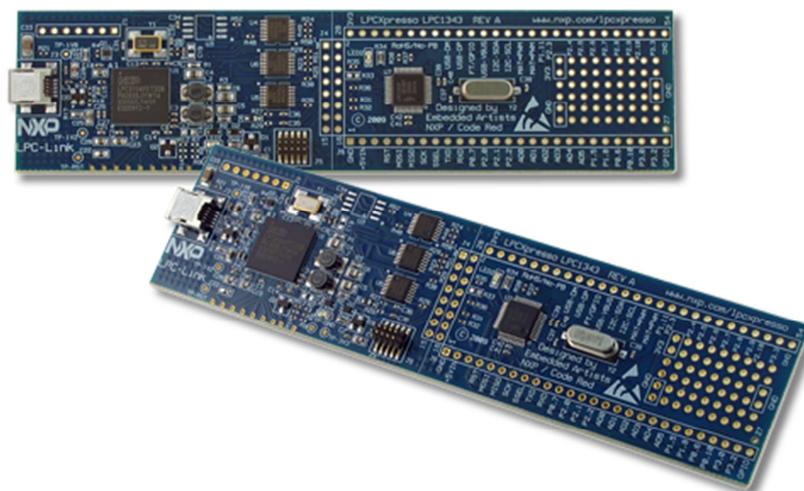


Figura 8 - Plataforma MBED.



**Figura 9 - Plataforma NXP LPCXpresso LPC1343.**

A Tabela 1 mostra a comparação das principais características dos microcontroladores analisados neste trabalho. As análises feitas a seguir mostram valores importantes quanto à escolha do microcontrolador para controle de um UAV quadrotor.

**Tabela 1 - Comparativo entre os microcontroladores ATmega1280, ATmega2560, LPC1343 e LPC1768.**

<b>Característica</b>	<b>ATmega1280</b>	<b>ATmega2560</b>	<b>LPC1343</b>	<b>LPC1768</b>
Frequência	16MHz	16MHz	72MHz	100MHz
Consumo	Não encontrado	Não encontrado	0,097 mW / MHz	0,097 mW / MHz
Desempenho (CoreMark/MHz)	0,53	0,53	2,17	2,17
Desempenho (DMIPS/MHz)	Não disponível	Não disponível	1,25	1,25
Estágios de <i>pipeline</i>	1	1	3	3
ADC	16 canais (10 bits)	16 canais (10 bits)	8 canais (10 bits)	8 canais (12 bits)
DAC	Não possui	Não possui	Não possui	1 canal (10 bits)
SRAM	8KB	8KB	8KB	64KB
EEPROM	4KB	4KB	Não possui	Não possui
<i>Flash</i>	Até 128KB	Até 256KB	Até 32KB	Até 512KB
GPIO	86	86	42	70
<i>Timer</i>	2 (8 bits) e 4 (16 bits)	2 (8 bits) e 4 (16 bits)	4 (32 bits)	4 (32 bits)
I2C	1	1	1	3
SPI	Sim	Sim	Sim	Sim
PWM	4 canais (8 bits) e 12 canais (2 a 16 bits)	4 canais (8 bits) e 12 canais (2 a 16 bits)	2 canais (32 bits) e 2 canais (16 bits)	6 canais (32 bits) e Controle de motor
USB	Não possui	Não possui	<i>Master/HID</i>	<i>Master/HID</i>
Preço ( <b>Digikey, 2011</b> )	US\$ 9,12 – US\$ 17,20	US\$ 10,15 – US\$ 19,05	US\$ 4,31 – US\$ 4,51	US\$ 11,44

Como pode ser visto, o microcontrolador LPC1343 possui um melhor custo benefício em relação ao ATmega2560, pois é mais barato e possui um maior desempenho, baseado no *benchmark* CoreMark, chegando a ser cerca de 18,5 vezes mais rápido. Então, se temos um microcontrolador que possui maior desempenho do que um microcontrolador que consegue processar as informações necessárias, então este também consegue. Em outras palavras, o microcontrolador

---

LPC1343 também pode ser utilizado para projeto de UAV quadrotor em que o ATmega2560 consegue suprir a necessidade de processamento.

Ainda assim, há uma questão importante. A maioria dos outros projetos utilizam Arduino que possui o microcontrolador ATmega2560 e possui toda sua simplicidade em prototipação de novos projetos, então tem de haver um bom motivo para se utilizar um microcontrolador ARM Cortex-M3. E há, esse motivo é a escalabilidade. Os microcontroladores da família Cortex-M3 possuem alta compatibilidade de código, assim caso seja necessário um microcontrolador com maior capacidade de processamento ou mais recursos, existe uma vasta gama onde pode ser escolhido um modelo que atenda aos requisitos necessários.

Os outros projetos citados são baseados na plataforma Arduino Mega 2560, que atualmente é a que possui o maior desempenho, assim um projeto de UAV quadrotor pode ser inviável de ser feito caso existam mais requisitos do que os que atualmente existem. Por exemplo, um requisito não especificado neste trabalho é o uso de visão computacional para reconhecimento dos objetos e/ou pessoas que se quer rastrear enquanto sobrevoa um determinado local. E a capacidade de memória e velocidade de processamento é o principal gargalo pra execução efetiva deste requisito.

Ainda há uma questão importante a ser analisada: Os projetos de UAV quadrotor analisados eram controlados por controle remoto ou tinham rota de voo fixa, mas neste trabalho há um principal requisito adicional em relação aos outros projetos: o microcontrolador tem que ter a capacidade de processar, em tempo real, o voo do UAV quadrotor em um enxame com processamento de rota dinâmica e coordenação na comunicação com os outros UAVs de forma assíncrona.

## 4.2 Magnetômetro

O magnetômetro é um instrumento de medição usado para medir a força ou direção de um campo magnético. Os magnetômetros utilizados em projetos deste tipo são dispositivos do tipo MEMS (do inglês - *Microelectromechanical systems*). O magnetômetro é utilizado em substituição à bússola comum, por conta da alta

---

sensibilidade a ruído e agrega funcionalidades adicionais. Alguns fabricantes desenvolvem circuitos integrados com sensoriamento de três eixos – nestes são utilizados três magnetômetros, cada um realiza o sensoriamento um diferente eixo e estes são encapsulados em um único circuito integrado – e possuem precisão que pode chegar entre 1º e 2º, com interface analógica ou digital.

Nos projetos (AeroQuad, 2011), (ArduCopter, 2011) e (Multipilot-32, 2011) é utilizado o HMC5883L do fabricante Honeywell, este possui interface I<sup>2</sup>C simples, baixo consumo de corrente e resolução de 5 mili-gauss (unidade de medição utilizada pelo magnetômetro para medir a direção e magnitude do campo magnético da Terra) (Honeywell, 2011).



Figura 10 - Placa de sensoriamento com magnetômetro HMC5883L utilizada o projeto AeroQuad.

No projeto (Multipilot-8, 2011) e no projeto desenvolvido por (Melo, 2010) não são utilizados magnetômetros, logo esses possuem menor precisão da sua posição em relação ao norte geográfico.

### 4.3 Acelerômetro

O acelerômetro é um dispositivo que mede a sua variação de velocidade em um dado sentido com relação a cada eixo. Os acelerômetros de três eixos informam as inclinações relacionadas aos eixos tridimensionais (XYZ). Os acelerômetros utilizados em projetos deste tipo também são do tipo MEMS.

---

O acelerômetro utilizado no projeto (AeroQuad, 2011) é o BMA180 do fabricante Bosch. Este é um acelerômetro de três eixos digital, de altíssimo desempenho e baixo consumo (tipicamente  $650\mu\text{A}$ ), com filtros digitais integrados programáveis (passa alta, passa baixa e passa faixa), saída digital com interface SPI e I2C, com faixa de sensoriamento programável (1g, 1,5g, 2g, 3g, 4g, 8g, 16g).



Figura 11 - Placa com acelerômetro BMA180 utilizado no projeto AeroQuad.

No projeto (ArduCopter, 2011) é utilizado o acelerômetro ADXL335 do fabricante Analog Devices. Este é um acelerômetro de três eixos, de baixo consumo, tipicamente  $350\mu\text{A}$ , que possui uma faixa de sensoriamento de  $\pm 3g$ .

No projeto desenvolvido por (Melo, 2010) foi utilizado o acelerômetro MMA7260Q do fabricante Freescale Semiconductor. Este é um acelerômetro de três eixos com faixa de sensoriamento programável (1,5g, 2g, 4g, 6g), baixo consumo (tipicamente  $500\mu\text{A}$ ).

Sobre os projetos (Multipilot-8, 2011) e (Multipilot-32, 2011) não foi encontrado modelo de acelerômetro utilizado, apenas descrição que era utilizado acelerômetro de três eixos.

## 4.4 Giroscópio

O giroscópio é um dispositivo de medição ou manutenção de orientação, baseado nos princípios de momento angular. Em essência, um giroscópio mecânico é uma roda giratória ou disco, cujo eixo é livre para tomar qualquer orientação.

---

Consiste de um rotor suspenso por um suporte formado por dois círculos articulados, vide Figura 12. Seu funcionamento baseia-se no princípio da força de Coriolis (Santos I. F., 2001). O eixo em rotação tem um efeito de memória que guarda direção fixa em relação ao círculo máximo dispensando as coordenadas geográficas. Este dispositivo informa a inclinação atual em cada eixo.



**Figura 12 - Princípio do giroscópio.**

No projeto (AeroQuad, 2011) foi utilizado o giroscópio ITG-3200 do fabricante InvenSense, vide Figura 13. Este é um giroscópio tri-axial do tipo MEMS, com filtro passa baixa programável digitalmente, sensor de temperatura, interface I2C, entrada de *clock* externo opcional e saída digital X-/Y-/Z- com escala completa de 2000°/s (InvenSense, ITG-3200, 2010).



**Figura 13 - Placa com giroscópio ITG-3200 utilizado no projeto AeroQuad.**

No projeto (ArduCopter, 2011) foram utilizados dois giroscópios do tipo MEMS: IDG-500 e ISZ-500, ambos do fabricante InvenSense. O primeiro é um giroscópio de dois eixos (X e Y) que possui amplificadores integrados, filtro passa baixa, sensor de temperatura, escala completa de 500°/s nas saídas X-/Y- e 110°/s nas saídas X/Y4.5 – o valor 4.5 significa a coeficiente de amplificação em relação ao sinal capturado pelo sensor e o fornecido para ser lido – , sensibilidade de 2,0mV/°s nas saídas X-/Y- e 9,1mV/°s nas saídas X/Y4.5 e saída analógica (InvenSense, IDG-500, 2008).

---

O segundo é um giroscópio de eixo único (Z) com filtro passa baixa, sensor de temperatura, escala completa de 500°/s na saída Z- e 110°/s na saída Z4.5, sensibilidade de 2,0mV/°s na saída Z- e 9,1mV/°s nas saídas Z4.5 e saída analógica (InvenSense, ISZ-500, 2009).

No projeto (Multipilot-8, 2011) não há referências sobre os giroscópios utilizados, apenas descrições que são três giroscópios MEMS com escala completa de 220°/s.

No projeto (Multipilot-32, 2011) não há referências sobre o giroscópio utilizado, apenas uma descrição que é um giroscópio tri-axial.

No projeto desenvolvido por (Melo, 2010) foram utilizados dois giroscópios o tipo MEMS: LPR510AL e LYS510LH, ambos do fabricante ST Microelectronics. O primeiro é um giroscópio MEMS biaxial (X e Y), com filtro passa baixa, escala completa de 400°/s e sensibilidade de 2,5mV/°s e saída analógica (STMicroelectronics, 2009). Sobre o segundo giroscópio, não foram encontrados documentos ou *datasheets* com as suas especificações.

## 4.5 Altímetro

O altímetro é um dispositivo que mede a altitude atual da aeronave. Existem altímetros barométricos, que medem a pressão atmosférica e calculam a altura atual de acordo com a pressão, assim sabe-se a altura em relação ao nível do mar. Como também baseados em radar, onde é emitida uma onda em direção ao solo e através do tempo passado para a onda ir e voltar é calculada a altura (Editora Abril, 2011).

No projeto (AeroQuad, 2011) é utilizado o sensor de pressão Bosch BMP085, vide Figura 14, que consegue medir de 300 a 1100 hPa (hectopascal – unidade de medida de pressão) com uma precisão absoluta de no mínimo 0,03 hPa, possui interface I2C e sensor de temperatura interno (Bosch, 2009).

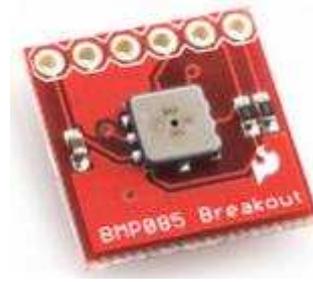


Figura 14 - Placa com o sensor de pressão Bosch BMP085 utilizado no projeto AeroQuad.

No projeto (ArduCopter, 2011) é utilizado, de forma opcional, o sonar MB1200 XL-MaxSonar-EZ0, que possui alcance de 20cm até 765cm com precisão de 1cm, e a leitura de dados pode ser feita digitalmente por largura de pulso ou serial, e de forma analógica.

No projeto (Multipilot-8, 2011) não há especificações do modelo de altímetro utilizado, apenas descrições de que são usados dois sensores de pressão com precisão de 20cm.

Sobre o projeto (Multipilot-32, 2011) não foram encontradas especificações do modelo de altímetro utilizado, apenas descrições que é um sensor de pressão do fabricante bosch.

No projeto desenvolvido por (Melo, 2010) não foi utilizado altímetro.

## 4.6 GPS

O GPS (*Global Positioning System*) é a abreviatura de NAVSTAR GPS (NAVSTAR GPS-NAVigation System with Time And Ranging Global Positioning System). É um sistema de radionavegação baseado em satélites desenvolvido e controlado pelo departamento de defesa dos Estados Unidos da América (U.S.DoD) que permite a qualquer usuário saber a sua localização, velocidade e tempo, 24 horas por dia, sob quaisquer condições atmosféricas e em qualquer ponto do globo terrestre (Malima, 2011).

O cálculo da posição é determinado a partir da determinação da distância entre o receptor e os satélites. Através das distâncias de pelo menos três satélites, podemos determinar a nossa posição relativa a esses 3 satélites através da intersecção de 3 circunferências cujos raios são as distancias medidas entre o

---

receptor e os satélites, como pode ser visto na Figura 15. Esse cálculo é também chamado de triangulação (Malima, 2011).

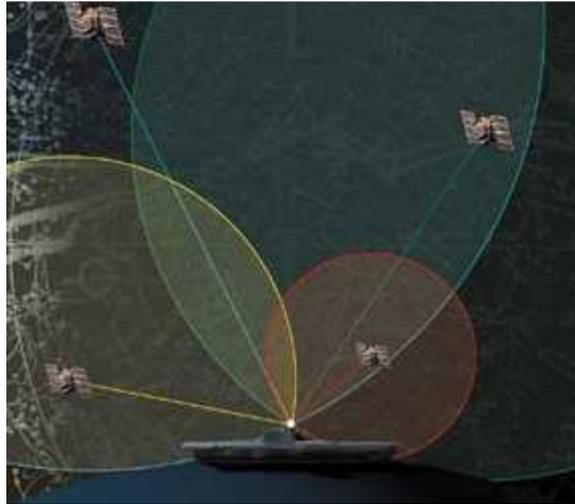


Figura 15 - Exemplo obtenção posicionamento baseado em triangulação.

No projeto (AeroQuad, 2011) não é utilizado GPS.

No projeto (ArduCopter, 2011) são propostos dois módulos GPS:

- GS407 U-Blox5, que possui chipset u-Blox 5H, 50 canais, taxa máxima de atualização de 4Hz, consumo de 75mA, sensibilidade de rastreamento menor que -160dBm, suporta função AGPS uBlox AssistNow e pesa 16g;
- MediaTek MT3329, que possui 66 canais, sensibilidade de rastreamento de até -165dBm, taxa de atualização de até 10Hz, precisão de posicionamento inferior a 3m, possui interface USB/UART, consumo de 48mA, suporta função AGPS e pesa 6g.

## 4.7 Motores

De maneira geral, motores são dispositivos que convertem energia, de diferentes tipos, em energia mecânica, na forma de força motriz para realização de algum trabalho (Halliday, Resnick, & Walker, 2009).

---

O foco de estudo deste trabalho será dado a motores elétricos de corrente contínua por ser o tipo de alimentação utilizado em projetos de UAV quadrotor de pequeno porte.

Os motores de corrente alternada aproveitam da própria alternância senoidal para energizar as bobinas e gerar torque no rotor. Porém os motores que recebem alimentação por corrente contínua devem ter uma forma para realizar a variação de corrente em cada bobina para gerar o movimento do rotor.

Existem dois tipos de motores DC, *brushed* (com escova) e *brushless* (sem escova), e em cada um há uma maneira de gerar rotação no rotor.

#### 4.7.1 Brushed

Os motores do tipo *brushed* (com escova) possuem comutadores eletromecânicos. Tais comutadores servem para que a corrente, que é conduzida para o rotor (armadura) por meio das escovas, troque de sentido fazendo que o campo magnético do rotor se inverta. Possuem o rotor magnetizado por indução, precisando de mecanismos que transfiram a energia elétrica para o rotor, sendo estes mecanismos as escovas e o comutador, vide Figura 16. O estator pode ser magnetizado por indução ou ter ímãs em seu conjunto (HowStuffWorks, 2011).

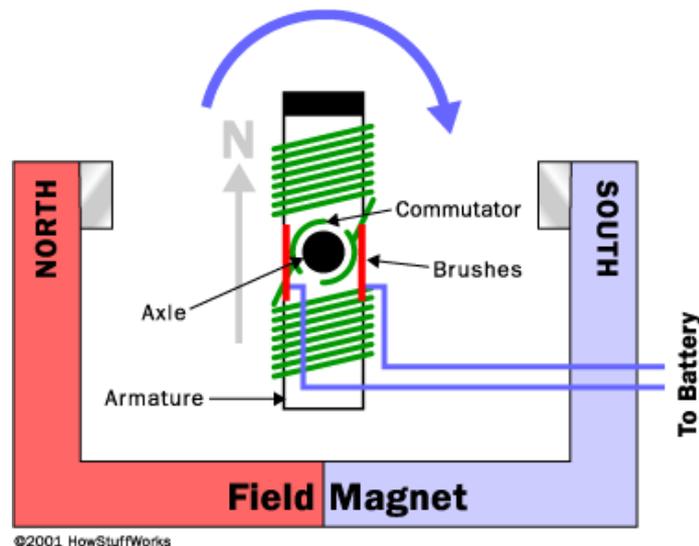


Figura 16 - Funcionamento de um motor de dois polos com escovas.

## 4.7.2 Brushless

Motores *brushless* possuem uma particularidade. Obrigatoriamente o rotor deve ser magnetizado por ímãs permanentes e não por indução. Já o estator é magnetizado pela corrente elétrica. Motores BLDC (*Brushless Direct Current*) possuem várias vantagens em relação aos motores com escova, dentre algumas estão:

- Melhor característica velocidade x torque;
- Resposta dinâmica alta;
- Alta eficiência;
- Maior vida útil;
- Silencioso;
- Altas taxas de velocidade;

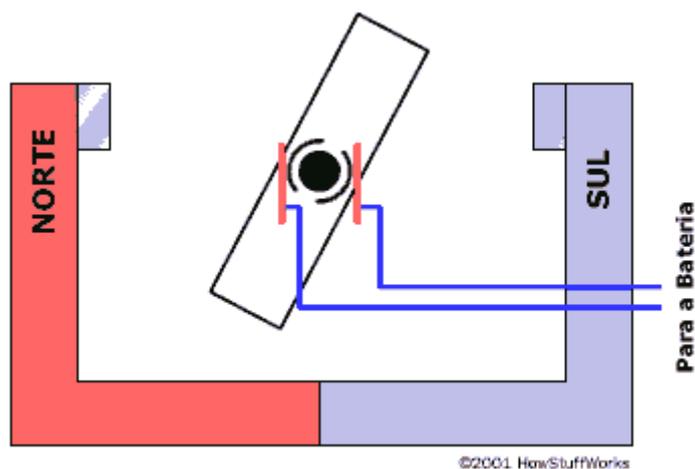


Figura 17 - Funcionamento de um motor de dois polos sem escovas.

Os motores BLDC podem ser de configuração em fase única, duas fases e três fases, que são os mais comuns. A comutação da corrente em cada fase é feita eletronicamente por um circuito de controle ligado ao motor (Microchip Technology, 2003). Estes circuitos são chamados de ESCs (do inglês – *Electronic Speed Controller*).

No projeto (AeroQuad, 2011) existem seis modelos que podem ser escolhidos:

- hexTronik DT750 Brushless Outrunner 750kv. Características: corrente máxima de 18A, 78g e 750rpm/v;

- 
- Turnigy TR2217/16 Outrunner. Características: corrente de máxima de 18A, 71g e 1050rpm/v;
  - BP A2217-9 Brushless Outrunner. Características: corrente máxima de 18A, 73,4g e 950rpm/v;
  - KEDA KDA20-22L hacker Style Brushless Outrunner. Características: corrente máxima de 17A, 56g e 924rpm/v;
  - hexTronik HXM2730-1300 Brushless Outrunner. Características: corrente máxima de 7,5A, 20g e 1300rpm/v.

No projeto (ArduCopter, 2011) existem dois modelos que podem ser escolhidos:

- AC2830-358. Características: corrente máxima de 12.2A, 62g e 850rpm/v;
- AC2836-358. Características: corrente máxima de 20A, 72g e 880rpm/v.

No projeto desenvolvido por (Melo, 2010) foi utilizado o motor E-max CF2822, que possui corrente máxima de 16A, 39g e 1200rpm/v.

Os projetos (Multipilot-8, 2011) e (Multipilot-32, 2011) são referentes apenas à parte de processamento de dados e comunicação.

Todos os motores citados são de três fases e sua alimentação pode ser feita através de uma bateria LiPo 3S, ou seja, 11,1V.

Admitindo-se alimentação através de uma bateria LiPo 3S e o uso de um ESC apropriado, o empuxo máximo que o motor consegue realizar depende apenas do motor e da hélice.

## 4.8 ESC

Os Controladores eletrônicos de velocidade ou ESCs (do inglês – *Electronic Speed Controllers*) são dispositivos que fornecem energia para alimentação dos motores e também realizam controle de velocidade dos motores em malha fechada. Os ESCs são circuitos eletrônicos, normalmente implementados com microcontrolador, que fornecem a alimentação correta para as três fases do motor BLDC. Essa alimentação é feita através de sinal PWM (do inglês – *Pulse Width Modulation*) com período de 20ms, e gerados sinais trapezoidais com deslocamento

de 120° em cada fase, vide Figura 18 e Figura 19. Há também ESCs que são controlados através de interface I2C, o que simplifica o controle, já que o barramento I2C suporta até 127 dispositivos sendo controlados com apenas dois pinos da interface I2C do microcontrolador. Os ESCs controlados por PWM precisam de um pino de saída pra cada ESC (Melo, 2010).

As saídas para alimentação das fases do motor (R, S e T) são mostradas na Figura 19, apresentadas por  $I_A$ ,  $I_B$  e  $I_C$ , respectivamente.

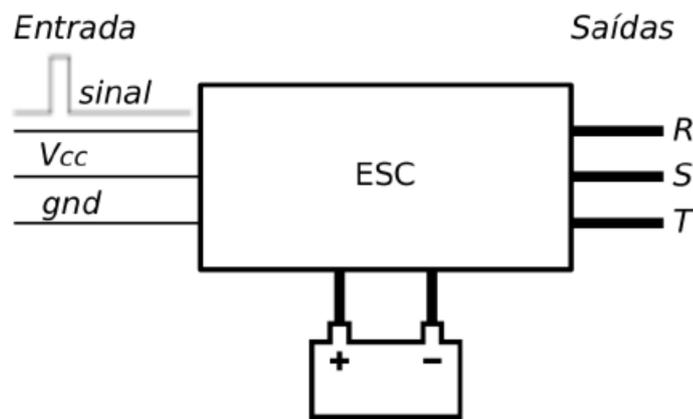


Figura 18 - Modelo de ESC com entrada de controle por PWM e três fases de saída para alimentação do motor BLDC.

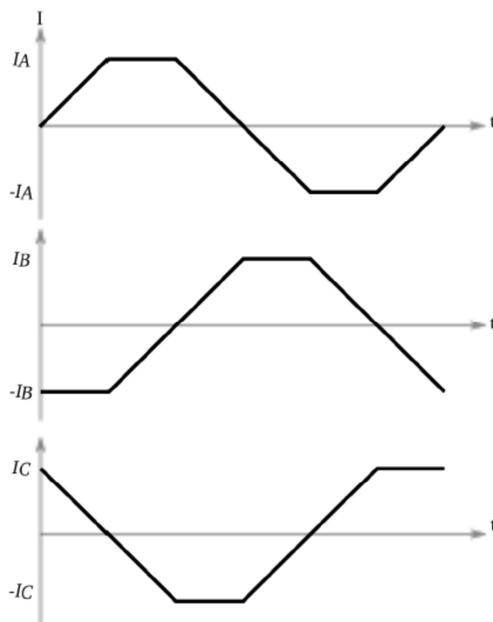


Figura 19 - Forma de onda gerada para alimentação de cada fase.

---

Para que suportem fornecer alimentação em grandes níveis de corrente, os ESCs utilizam transistores do tipo MOSFET, dois para cada fase.

No projeto (AeroQuad, 2011) os ESCs recomendados são:

- TURNIGY Plush 10amp 9gram Speed Controller, que fornece 10A de corrente continuamente, até 12A em pico, pesa 9g e velocidade máxima para motores BLDC de 21000rpm (2 polos), 7000rpm (6 polos) e 3500rpm (12 polos).
- TURNIGY Plush 18amp Speed Controller, que fornece 18A de corrente continuamente, até 22A em pico, pesa 19g e velocidade máxima para motores BLDC de 21000rpm (2 polos), 7000rpm (6 polos) e 3500rpm (12 polos).
- TURNIGY Plush 25amp Speed Controller, que fornece 25A de corrente continuamente, até 35A em pico, pesa 22g e velocidade máxima para motores BLDC de 21000rpm (2 polos), 7000rpm (6 polos) e 3500rpm (12 polos).
- HobbyWing FlyFun Brushless ESC 30A, que fornece 30A de corrente continuamente, até 40A em pico por 10 segundos, pesa 25g e velocidade máxima para motores BLDC de 21000rpm (2 polos), 7000rpm (6 polos) e 3500rpm (12 polos).
- HobbyWing FlyFun Brushless ESC 18A, que fornece 18A de corrente continuamente, até 22A em pico por até 10 segundos, pesa 18g e velocidade máxima para motores BLDC de 21000rpm (2 polos), 7000rpm (6 polos) e 3500rpm (12 polos).

Todos os ESCs sugeridos no projeto AeroQuad são controlados através de PWM.

No projeto (ArduCopter, 2011) não foi especificado o modelo dos ESCs utilizados, apenas existem informações que são de 20A.

Os projetos (Multipilot-8, 2011) e (Multipilot-32, 2011) são referentes apenas à parte de processamento de dados e comunicação.

---

No projeto desenvolvido por (Melo, 2010) foi utilizado um E-MAX ESC-18A, que fornece 18A de corrente continuamente, até 23A em pico, pesa 22g e velocidade máxima para motores BLDC de 24000rpm (2 polos), 8000rpm (6 polos) e 4000rpm (12 polos).

## 4.9 Bateria

Uma característica importante para escolha dos componentes descritos anteriormente é a capacidade de fornecimento de energia, pois está diretamente ligada ao tempo de voo do UAV.

Existem diversos tipos de baterias com diferentes características e capacidades, porém em alguns dos tipos de motores, como mostrado na seção 4.6, consomem uma corrente relativamente alta em comparação com sistemas embarcados de controle. Por exemplo, existem ESCs para esta aplicação que consomem 30A (AeroQuad-Parts-List, 2011) e o microcontrolador LPC1768 utilizado no projeto (MBED, 2011), com todas as suas funcionalidades, consome no máximo 500mA (NXP-LPC17XX, 2010), mostrando ser uma diferença muito significativa.

A seguir, na Tabela 2 são mostrados os mais comuns tipos de bateria utilizados atualmente e a relação de capacidade, volume e massa.

**Tabela 2 - Relação dos tipos de bateria - capacidade, volume e massa.**

Material	Relação Capacidade/Volume	Relação Capacidade/Massa
Li-Ion Battery ( <b>Electronic-Labs, 2011</b> )	400 Wh/l	160 Wh/kg
NiMH Battery ( <b>Everything2, 2011</b> )	280 Wh/l	100 Wh/kg
Lead-Acid Battery ( <b>Everything2, 2011</b> )	40 Wh/l	25 Wh/kg
LiPo Battery ( <b>Wikipedia, Lithium-ion_polymer_battery, 2011</b> )	300 Wh/l	130-200 Wh/Kg

Nos projetos (AeroQuad, 2011), (ArduCopter, 2011), (Multipilot-8, 2011), (Multipilot-32, 2011) e no projeto desenvolvido por (Melo, 2010) são utilizadas baterias do tipo LiPo (do inglês - *Lithium-ion polymer*), que é uma bateria

---

recarregável que possui características propícias para este tipo de aplicação, pois não possui o efeito de memória, o que lhe dá uma maior vida útil, possui também uma baixa auto descarga, aproximadamente 5% ao mês, e química melhorada, que lhe dá uma maior densidade de carga, alta tensão em malha aberta (3,6V a 3,7V por célula) e tem capacidade de fornecer alta corrente de descarga (Moore, 2008).

Os fabricantes utilizam representações específicas para as características das baterias. Uma dessas denominações está relacionada ao esquema de ligação das células que é no formato **xSyP**, que significa que esta bateria possui um grupo de **x** células em série (S) e **y** grupos em paralelo (P). Logo, sabe-se a tensão pela quantidade de células em série e sabe-se a quantidade de carga pela quantidade de grupos em paralelo, por exemplo uma bateria 3S2P com células de 3,7V e 2500mAh cada, possui 11,1V de tensão e capacidade de 5000mAh. Outra denominação é sobre a quantidade máxima de corrente que pode ser fornecida pela bateria que é no formato **xC**, que é a taxa de 1000:1 da capacidade da bateria em mAh em relação a corrente máxima que pode ser fornecida, por exemplo uma bateria 20C com carga de 5000mAh pode fornecer uma corrente de até 100A continuamente (Leptron, 2011).

No projeto (AeroQuad, 2011) foi utilizada a bateria de LiPo: Zippy Flightmax 3S1P 20C que possui 4000mAh de carga, 11,1V de tensão e massa de 306g. No projeto desenvolvido por (Melo, 2010) foi utilizada a bateria de LiPo: Thunder Power Pro Power 3S 30C que possui 2250mAh de carga, tensão de 11,1V e massa de 170g. Sobre os projetos (ArduCopter, 2011), (Multipilot-8, 2011) e (Multipilot-32, 2011) não foram encontradas especificações da bateria utilizada, apenas descrição que uma é bateria do tipo LiPo.

Em todos os projetos citados, foi utilizada bateria LiPo 3S (3 células em série, o que resulta uma tensão de 11,1V) e, das que as especificações foram encontradas, a diferença mais relevante entre as baterias eram os valores de carga e o peso, percebe-se então que a bateria a ser utilizada em projetos deste tipo deve ser do tipo LiPo, mas ainda deve-se ter atenção na escolha da bateria por conta da capacidade de carga, já que a quantidade de carga é quase que proporcional ao tempo de voo, por conta do consumo de energia dos motores. E uma bateria com

---

maior capacidade é mais pesada. Assim ao escolher uma bateria com maior capacidade, aumenta-se também a massa total do UAV, que pode levar a necessidade de motores mais potentes, que, por conseguinte gastam mais energia. Por fim, nota-se que a capacidade da bateria e a potência dos motores são grandezas conflitantes em relação ao tempo de voo, e deve-se ter cuidado na escolha destes dois dispositivos.

## 4.10 Comunicação sem fio de longo alcance

Para comunicação de entre os UAVs quadrotor é necessário que se tenha dispositivos de comunicação de longo alcance, pois os UAVs, quando implementado o enxame para realização de monitoramento, devem patrulhar a maior área possível.

Analisando requisitos como alcance, consumo de energia e topologia de rede, percebe-se que um padrão que atende bem a esses requisitos é o padrão IEEE 802.15.4 (IEEE, 2011). Este padrão especifica a camada física e MAC para baixas taxas de transferência em WPANs (do inglês - *Wireless Personal Area Networks*). Este padrão possui transferência de dados de até 250Kbps, que para as trocas de informações entre os UAVs podem ser suficientes, baixa latência, para respostas rápidas, estabelecimento automático da rede realizada por um coordenador, gerenciamento de energia para garantir baixo consumo, 16 canais na banda ISM 2.4Ghz, 10 canais em 915Mhz e um canal na banda 868Mhz (IEEE, 2011).

Sua topologia define dois tipos de nó: Dispositivos de funções completas (FDD, do inglês - *Full-Function Device*) e dispositivos de funções reduzidas (RDD, do inglês - *Reduced-Function Devices*). Os dispositivos de funções completas podem atuar como o coordenador da rede e implementam um modelo que permitem se comunicar com qualquer outro dispositivo. Os dispositivos de funções reduzidas são extremamente simples e podem apenas se comunicar com os coordenadores, e nunca podem ser coordenadores. Como pode ser visto na Figura 20, podem ser formadas duas topologia: em estrela ou ponto-a-ponto, e para tanto precisam de pelo menos um dispositivo FDD para realizar a função de coordenador (Santos S. T., 2007).

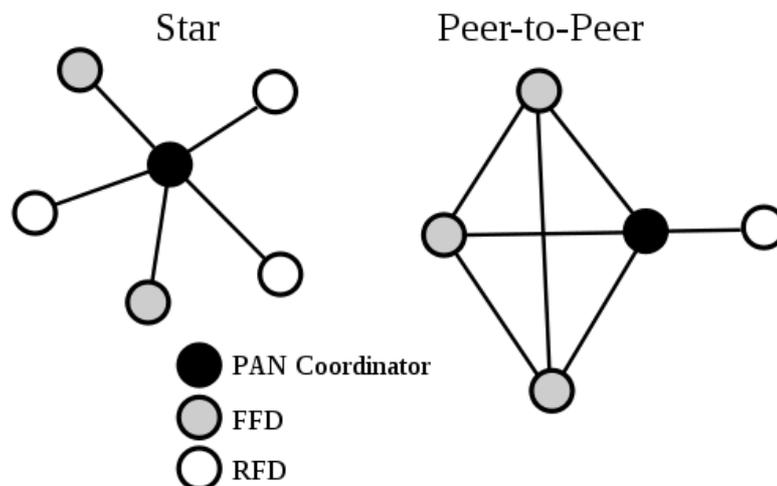


Figura 20 - Topologias de rede do padrão IEEE 802.15.4.

Com base neste padrão há algumas especificações que são desenvolvidos dispositivos de comunicação. Estes dispositivos utilizam a camada física e MAC já definida e implementam as camadas superiores. O mais comum e fácil de encontrar no mercado é o ZigBee (Santos S. T., 2007).

Nos projetos (AeroQuad, 2011), (Multipilot-8, 2011), (Multipilot-32, 2011) e no projeto desenvolvido por (Melo, 2010) não são utilizados módulos de comunicação baseados no IEEE 802.15.4, são utilizados rádios receptores, pois o controle é realizado através um controle remoto de aeromodelismo.

No projeto (ArduCopter, 2011) foi utilizado o módulo XBee-PRO 900 extended range, que possui consumo de 210mA, taxa de transferência de 156kbps, potência de transmissão de 50mW, até 10Km de alcance quando utilizado com antenas de alto ganho, suporta redes grandes e densas, opera em 900Mhz, possui criptografia AES de 128 bits e suporte a avançadas redes em malha e modos de baixo consumo de energia.

---

# Capítulo 5

## Experimento controlado para validação do desempenho da plataforma Arduino Mega

Neste capítulo será mostrado um experimento controlador para verificação e validação da capacidade do microcontrolador Atmel ATmega1280 em realizar o controle dos dispositivos utilizados em UAV quadrotor para aplicação em enxame de UAVs.

Nos projetos (AeroQuad, 2011) e (ArduCopter, 2011) é utilizada a plataforma Arduino Mega para processamento dos dados e controle dos motores do UAV quadrotor, porém nestes projetos o controle do quadrotor é realizado por uma pessoa através de um controle remoto sem fio.

Este experimento quer descobrir se a plataforma Arduino Mega, já utilizada em outros projetos, consegue processar todas as informações dos sensores e realizar o controle dos motores do UAV quadrotor como já faz e, além disso, processar os dados para controle de voo automático com rota dinâmica e coordenação de voo entre os outros UAVs, baseado em PSO (do inglês – *Particle Swarm Optimization*) (Kennedy & Eberhart, 2005).

A plataforma Arduino Mega possui o microcontrolador Atmel ATmega1280 embarcado e a plataforma Arduino Mega2560 possui o microcontrolador Atmel ATmega2560. A única característica diferente entre esses microcontroladores é a quantidade de memória *flash*, onde o ATmega1280 possui 128KB e o ATmega2560 possui 256KB. Esta diferença apenas impacta em projetos grandes, onde o código compilado é maior que 128KB de memória, dessa forma o ATmega1280 não pode ser utilizado, porém em outros casos não há diferença na utilização de qualquer um destes dois microcontroladores.

---

## 5.1 Microcontrolador Atmel ATmega1280

Este microcontrolador é um dos microcontroladores da arquitetura Atmel MegaAVR de 8bits. O ATmega1280 possui arquitetura RISC com 135 instruções, 32 x 8 registradores de propósitos gerais, multiplicador com operação em dois ciclos, até 128KB de *flash*, 4KB de memória EEPROM, 8KB de memória SRAM, entre outros (Atmel, 2011). Algumas informações sobre este microcontrolador podem ser vistas na Tabela 1.

## 5.2 Particle Swarm Optimization

Otimização por Enxame de Partículas ou PSO (do inglês - *Particle Swarm Optimization*), é uma técnica de otimização estocástica de funções não-lineares baseada em populações. Criado com a inspiração biológica na simulação do comportamento social de um bando de pássaros, proposto por (Kennedy & Eberhart, 2005). Sua concepção é feita a partir de um conceito muito básico com regras simples e fáceis de implementar, e requer apenas alguns operadores matemáticos primitivos e poucos recursos de memória e processamento (Kennedy & Eberhart, 2005).

No PSO, cada partícula, ou seja, cada indivíduo representa uma possível solução para o problema. As partículas se movimentam no espaço de busca em forma de “bando” com o objetivo buscar a solução ótima. A mudança de posição de cada partícula no espaço de busca é baseada na tendência de cada indivíduo em voltar à sua melhor posição (componente cognitivo) e da tendência em ir para a melhor posição da sua vizinhança (componente social) (Kennedy & Eberhart, 2005).

Este algoritmo foi utilizado para estimar o custo computacional o algoritmo de coordenação de enxame proposto por (Pinheiro, 2011), que se baseia no PSO para coordenar um enxame de UAVs.

---

## 5.3 Objetivos

### 5.3.1 Objetivo global

Analisar o desempenho da plataforma Arduino Mega com o propósito de determinar sua capacidade de processar as informações dos sensores, a comunicação e o controle dos motores, baseado nos algoritmos utilizados no projeto (AeroQuad, 2011), e processar os dados para controle de voo automático com rota dinâmica e coordenação de voo entre os outros UAVs, simultaneamente, em relação ao tempo de processamento no contexto de sistemas embarcados para controle de UAVs.

### 5.3.2 Objetivos do estudo

Analisar o desempenho da plataforma Arduino Mega com o propósito de validar a capacidade de execução dos algoritmos de controle dos sensores, da comunicação e dos e motores do quadricóptero UAV, com respeito à comparação do uso ou não uso do algoritmo de controle de enxame de UAVs, baseado em PSO, do ponto de vista do tempo de execução.

### 5.3.3 Questão:

A plataforma Arduino Mega é capaz de executar os algoritmos de controle do UAV e algoritmo de controle do enxame de UAVs, simultaneamente, sem impactar no desempenho do voo do UAV quadrotor?

## 5.4 Planejamento

### 5.4.1 Definição das hipóteses

**Hipótese Nula ( $H_0$ ):** A plataforma Arduino Mega não consegue executar todos os algoritmos simultaneamente

$$H_0: \mu_{F1-A1} - \mu_{F1-A2} > \theta \quad (1).$$

**Hipótese Alternativa ( $H_1$ ):** A plataforma Arduino Mega consegue executar todos os algoritmos simultaneamente

$$H_1: \mu_{F1-A1} - \mu_{F1-A2} \leq \theta \quad (2).$$

---

Onde:

$\mu_{F1-A1}$  é a média das amostras dos tempos de execução dos algoritmos do projeto AeroQuad junto com PSO.

$\mu_{F1-A2}$  é a média das amostras dos tempos de execução dos algoritmos do projeto AeroQuad sem o PSO.

$\theta$  é o limiar para a diferença das médias das amostras dos tempos de execução.

#### **5.4.2 Descrição da instrumentação**

Como instrumentos deste experimento são utilizados:

- Placa Arduino Mega que possui o microcontrolador ATmega1280;
- IDE Arduino Alpha versão 0022.

#### **5.4.3 Seleção do contexto**

O contexto caracteriza-se em quatro dimensões:

- Processo: On-line;
- Os participantes: Estudantes de engenharia de computação - UPE;
- Realidade: Problema real;
- Generalidade: específico.

Este estudo propõe um processo on-line, pois a plataforma não sofrerá nenhum tipo de modificação em sua estrutura futura.

Os participantes são estudantes de engenharia de computação da Universidade de Pernambuco: David Alain, Rafael Farias e Caio Bernardes.

O estudo é um problema real por causa das análises serem feitas na plataforma Arduino Mega, já que esta é utilizada em projetos de UAV quadrotor.

Os tempos de execução são comparados em relação a algoritmos que são executados no mesma plataforma, então o contexto possui caráter específico.

---

#### 5.4.4 Variáveis

##### Independentes

- Uso e não uso do algoritmo de controle de enxames de UAVs baseado em PSO.
  - Alternativa1 (F1-A1): o PSO é executado junto com os algoritmos do projeto AeroQuad.
  - Alternativa2 (F1-A2): o PSO não executado e são executados apenas os algoritmos do projeto AeroQuad.

##### De controle

- Arduino Mega:

Diferentes plataformas possuem características que resultam em níveis de desempenho diversos. Para este experimento, será considerada apenas essa plataforma.
- Algoritmo PSO:

O PSO utilizado foi implementado em linguagem C++. São utilizadas apenas três partículas, já que na proposta de (Pinheiro, 2011), um UAV é representado por uma partícula e usa o conceito de comunicação 2-conectado para topologia de vizinhança da partícula. Dessa forma cada partícula só se comunica com as duas mais próximas. A implementação pode ser vista no Apêndice A.

Como o UAV é representado por uma partícula, só precisa ser realizada a atualização da posição e velocidade de uma única partícula, a que representa a si mesmo.

Foi admitido que a posição dos dois vizinhos será recebida através da comunicação entre os UAVs, logo o PSO implementado não se encarregará sobre a captura dos dados, apenas da leitura dos seus valores.
- Algoritmos de controle dos sensores, da comunicação e dos motores:

---

O experimento utiliza os algoritmos em C/C++ do projeto AeroQuad, disponível em (AeroQuad\_Sources, 2011), para ambos os tratamentos considerando que esses algoritmos exercerão a mesma influência na variável de resposta.

### **Dependentes**

- Tempo de execução:

Essa variável será medida utilizando uma escala razão.

#### **5.4.5 Análise quantitativa**

Para analisar os tempos de execução resultantes das simulações do uso dos algoritmos, se propõe aplicar a análise quantitativa. Esta análise deve ser capaz de comparar o tempo de execução das simulações de duas combinações de uso dos algoritmos envolvidos.

#### **5.4.6 Tipo de projeto**

Um fator, dois tratamentos, de comparação simples.

#### **5.4.7 Coleta, validação e armazenamento de dados**

Os dados de contagem de tempo de execução, em microssegundos, serão coletados para cada simulação feita e armazenados em planilhas eletrônicas com formato em coluna. Admite-se que todos os dados extraídos nessa etapa são válidos, caso não haja casos de cancelamento ou eventual erro durante execução.

#### **5.4.8 Análise dos dados**

Como teste estatístico de hipótese será utilizado o teste-t de Student para cálculo do intervalo de confiança da diferença entre médias com um nível de significância de 1%, ou seja, 99% de certeza. Devido à hipótese alternativa considerar a diferença dos tempos de execução, o teste será unicaudal e independente. Para isso, se utilizará a fórmula:

$$\mu_{F1-A1} - \mu_{F1-A2} \pm Z_c \sqrt{\frac{\sigma_{F1-A1}^2}{N_{F1-A1}} + \frac{\sigma_{F1-A2}^2}{N_{F1-A2}}} \quad (3).$$

---

Onde:

$\mu_{F1-A1}$  é média dos tempos de execução dos algoritmos com PSO;

$\sigma_{F1-A1}^2$  é o desvio padrão dos tempos de execução dos algoritmos com PSO;

$\mu_{F1-A2}$  é a média dos tempos de execução dos algoritmos sem PSO;

$\sigma_{F1-A2}^2$  é o desvio padrão dos tempos de execução sem PSO;

$N_{F1-A1}$  é a quantidade de simulações com PSO;

$N_{F1-A2}$  é a quantidade de simulações sem PSO; e

$Z_C$  é valor da tabela T de Student para um dado intervalo de confiança.

## 5.5 Execução

Em cada simulação foram executadas 50 iterações do *loop* principal e o tempo total da execução será dividido por 50, pegando um valor médio do tempo de execução de cada iteração.

Este *loop* tem uma característica específica onde se deve ter cuidado. Cada dispositivo, quando definido pra ser utilizado, tem uma frequência diferente para leitura ou atualização dos dados:

- Giroscópio a 100Hz;
- Acelerômetro a 100Hz;
- DCM (DIY-Drones, 2011) a 100Hz;
- Telemetria (transmissor de *status* através de ZigBee ou Bluetooth) a 10Hz;
- Controle dos motores a 100Hz;
- *Receiver* (comunicador do controle remoto) a 50Hz;
- Altímetro a 25Hz;
- Magnetômetro a 10Hz;
- Monitor de bateria a 10Hz.

O microcontrolador consegue executar o *loop* numa frequência superior a 100Hz, que é maior entre para atualização dos dados dos dispositivo. Nota-se então que pode acontecer de não serem executados algoritmos de alguns dos dispositivos, dado que naquele instante pode não haver completado ainda o período para que seja executado novamente. Então a contagem do tempo de desta forma de execução poderia gerar dados inválidos.

---

Para tanto, esses algoritmos foram executado na velocidade do microcontrolador sem o controle de frequência de atualização, que é o pior caso em relação ao tempo de execução do *loop*, quando todos os algoritmos são executados. Os dados extraídos desta maneira são válidos, pois o que se quer é a diferença do tempo de execução do *loop* quando executado com e sem o algoritmo PSO.

Executando a simulação dos algoritmos do projeto AeroQuad sem incluir o PSO, obtemos os tempos, onde calculamos a média e desvio padrão da amostra. O limiar  $\theta$  será definido baseando nessas amostras.

Para realizar as amostragens, sem que houvesse viés, antes de realizar cada simulação o microcontrolador era desligado e ligado. No momento que era ligado, o microcontrolador inicializava todas as variáveis e estados, configurava os dispositivos e iterava o *loop* principal por 50 vezes.

O tempo de execução da inicialização e configuração dos dispositivos não é interessante neste projeto já que isto só acontecerá uma vez, ao ligar o UAV, e não interfere na capacidade de voo do UAV, pois é executado antes do UAV decolar.

## 5.6 Resultados obtidos

Após execução dos experimentos, os dados foram extraídos e adicionados a uma planilha eletrônica para serem feitas as análises. Estes dados estão na Tabela 3 a seguir.

Tabela 3 - Resultados das simulações sem a utilização do PSO.

Sem PSO		
4838,96	4645,04	4842,96
4846,88	4865,12	4830,08
4846,16	4871,68	4839,84
4863,76	4874,96	4818
4858,32	4871,12	4844,08
4694,96	4861,44	4850,16
4646,08	4682,8	4848
4643,92	4838,48	4846,56
4650,32	4839,52	4839,76
4674,56	4837,12	4841,92

Os valores na Tabela 3 são os tempos de execução em microssegundo, para cada simulação. Foram colocados em três colunas para fins de organização. Para uma visualização mais simples dos dados, foi plotado um gráfico, vide Figura 21, com os dados extraídos das simulações executadas. No eixo vertical está a indicação do tempo de execução em microssegundos e no eixo horizontal está a indicação da simulação.

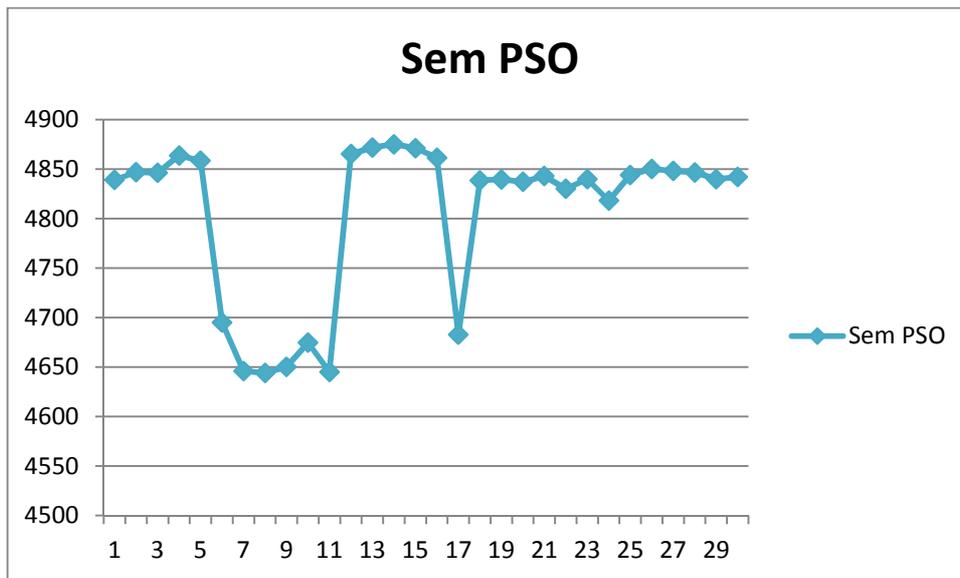


Figura 21 - Tempos de execução das amostras de simulações sem a utilização do PSO.

Realizando análises nesses dados obtemos a média e desvio padrão das amostras. Onde a média obtida foi  $\mu_{F1-A2} = 4805,085333\mu s$  e o desvio padrão obtido foi  $\sigma_{F1-A2}^2 = 81,51416631\mu s$ . Assim, estimando o pior caso para o tempo de execução sem o PSO como sendo:

$$pior_{F1-A2} = \mu_{F1-A2} + \sigma_{F1-A2}^2 = 4886,5995\mu s,$$

e conseguimos fazer uma estimativa para o limiar  $\theta$ .

Sabe-se que a frequência de atualização dos dados é de 100Hz (período de 10ms), podemos estimar o limiar  $\theta$  através do tempo máximo de execução  $\tau = 10000\mu s$  e o pior caso  $pior_{F1-A2} = 4886,5995\mu s$ . Então este limiar será definido em:

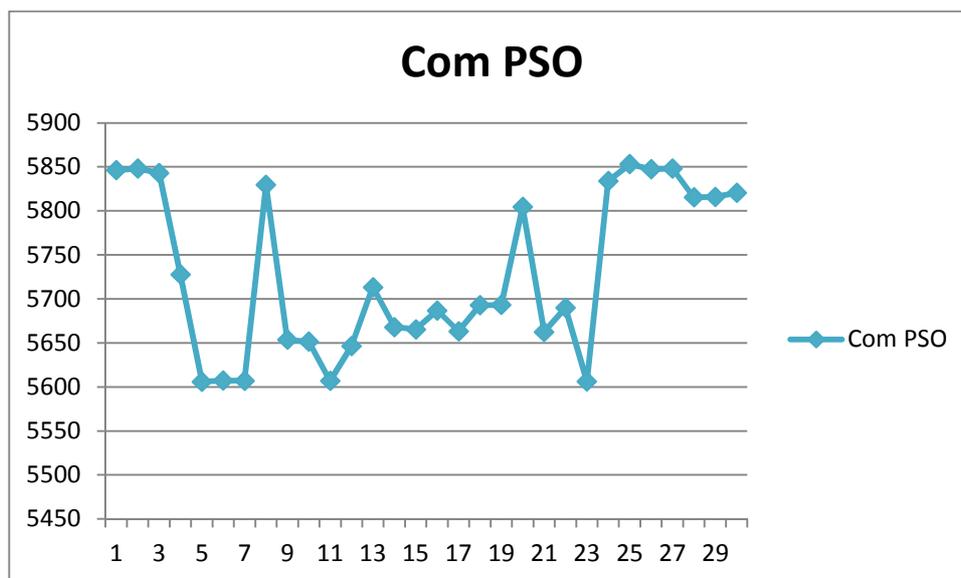
$$\theta = \tau - pior_{F1-A2} = 5113,4005\mu s.$$

Na execução das simulações dos algoritmos junto com o PSO foram obtidos os valores mostrados na Tabela 4.

**Tabela 4 - Resultados das simulações com a utilização do PSO.**

Com PSO		
5846,32	5606,88	5662,64
5848,16	5646,56	5689,92
5843,2	5713,36	5606,24
5727,6	5667,92	5833,92
5606	5665,44	5853,2
5607,2	5686,8	5847,28
5607,04	5663,36	5848,16
5829,6	5692,8	5815,44
5653,6	5693,28	5816
5651,44	5804,48	5820,56

Os valores na Tabela 4 são os tempos de execução em microssegundo, para cada simulação. Foram colocados em três colunas para fins de organização. Foi plotado outro gráfico para visualização dos dados, vide Figura 22, com os dados extraídos das simulações executadas com o PSO. No eixo vertical está a indicação do tempo de execução em microssegundos e no eixo horizontal está a indicação da simulação.



**Figura 22- Tempos de execução das amostras de simulações com a utilização do PSO.**

---

### 5.6.1 Validação das hipóteses

Para validação das hipóteses iremos tentar refutar a hipótese nula  $H_0$ , para provar que a plataforma Arduino Mega consegue executar todos os algoritmos, incluindo o PSO. Caso a hipótese nula não seja refutada, não podemos afirmar que o microcontrolador consegue executar os algoritmos.

Os dados obtidos foram:

$$\mu_{F1-A1} = 5728,48\mu s \text{ e } \sigma_{F1-A1}^2 = 92,99688508\mu s;$$

$$\mu_{F1-A2} = 4805,085333\mu s \text{ e } \sigma_{F1-A2}^2 = 81,51416631\mu s;$$

$$\theta = 5113,4005\mu s.$$

Ao analisar o teste-T para o intervalo de confiança da diferença entre duas amostras obtemos:

$$5728,48 - 4805,085333 \pm 2,58 \sqrt{\frac{92,99688508}{30} + \frac{81,51416631}{30}}$$

Que é igual a

$$923,394667 \pm 4,252806.$$

Como o limiar  $\theta = 5113,4005$  e a diferença obtida é  $923,394667 \pm 4,252806$ , prova-se que a diferença é menor ou igual ao limiar, então a hipótese nula é refutada com 99% de certeza. Logo a hipótese alternativa é verdadeira provando que o microcontrolador plataforma Arduino Mega consegue executar todos os algoritmos do projeto AeroQuad e o PSO simultaneamente, sem que se tenha atrasos.

Assim, pode-se afirmar que a plataforma Arduino Mega pode ser utilizada como unidade de processamento em UAVs que realizam voo com rota dinâmica implementados em enxame de UAVs.

---

# Capítulo 6

## Resultados

Após os estudos realizados sobre os dispositivos necessários a construção de UAVs quadrotor obteve-se dados e conhecimento suficientes para a proposição de quais dispositivos devem ser utilizados.

A proposição dos dispositivos se baseia nas funcionalidades de cada dispositivo e aqui serão descritos os mais indicados para o projeto de UAV quadrotor cujo foco é implementação de enxame de UAVs para realização de monitoramento territorial.

### 6.1 Dispositivos propostos

#### 6.1.1 Microcontrolador

NXP LPC1768, visto que este tem capacidade de processamento muito superior ao microcontrolador utilizado na plataforma Arduino Mega, e também possui capacidade processar algoritmos de visão computacional (Rufino, 2011).

#### 6.1.2 Magnetômetro

Honeywell HMC5883L, visto que este possui saída de dados através de interface I2C, o que simplifica o projeto por não precisar utilizar conversores AD do microcontrolador e como podem ser utilizados vários dispositivos no mesmo barramento I2C, o projeto da placa que contenham os sensores será mais simples.

#### 6.1.3 Acelerômetro

Bosch BMA180, pois este possui saída de dados através de interface I2C e é o mais preciso dentre os analisados neste trabalho.

---

#### **6.1.4 Giroscópio**

InvenSense ITG-3200, pois este também possui saída de dados através de interface I2C e também é o mais preciso dentre os analisados neste trabalho.

#### **6.1.5 Altímetro**

Bosch BMP085, pois é bem preciso e possui interface I2C.

#### **6.1.6 GPS**

MediaTek MT3329, pois mais canais, melhor sensibilidade de rastreamento, taxa de atualização de posicionamento mais alta, menor consumo de energia e menor peso.

#### **6.1.7 Motores**

Turnigy TR2217/16, pois a taxa rpm/V é uma das melhores, se comparando com os outros, e a relação peso/potência dele é um pouco melhor que os outros.

#### **6.1.8 ESCs**

E-MAX ESC-18A, pois atende as especificações do motor escolhido e dentre os ESCs estudados, que possui corrente de 18A é o que tem motores mais velozes, logo possui força de empuxo maior.

#### **6.1.9 Bateria**

Thunder Power Pro Power 2250mAh, pois esta possui massa de apenas 170g.

#### **6.1.10 Comunicação sem fio de longo alcance**

XBee-PRO 900 extended range, por todas as características descritas anteriormente.

---

# Capítulo 7

## Conclusão e Trabalhos Futuros

Após os estudos realizados neste trabalho, pode-se concluir que a escolha dos dispositivos a serem utilizados em um projeto de UAV quadrotor devem ser feitas para cumprir com os requisitos do problema que o UAV quadrotor vai resolver.

Os requisitos de processamento de dados pode mudar microcontrolador a ser utilizado, ou até requerer um microprocessador para processamento desses dados.

A escolha dos sensores, como giroscópio, acelerômetro, magnetômetro, altímetro e GPS, vai impactar na precisão das informações, logo se deve analisar a precisão necessária ao projeto para que se tenha efetividade nas tomadas de decisão de acordo com o estado atual do UAV.

Para escolha dos motores deve ser analisada a massa total do quadrotor, onde uma característica importante a se analisar sobre os motores é a força de empuxo. Logo, a massa total do quadrotor, vai indicar os motores e hélices necessários.

Para controle dos motores são utilizados ESCs, e estes são escolhidos de acordo com as especificações de potência dos motores a serem utilizados. Os ESCs devem possuir capacidade de potência superiores às especificações dos motores, para que estes não tenham sua vida útil diminuída sem que ocorra sobre aquecimento dos seus componentes, o que acarreta em um maior consumo de energia.

As baterias a serem utilizadas devem possuir especificações de corrente necessitada pelos ESCs, já que os ESCs controlam a potência dos motores, como também. A bateria deve ser escolhida, em relação a sua capacidade de carga, deve ser escolhida de acordo com o tempo que se quer que o UAV fique voando ininterruptamente, porém uma bateria com maior carga é mais pesada e demanda motores mais potentes. E assim percebe-se que a capacidade da bateria e a potência dos motores são grandezas conflitantes em relação ao tempo de voo.

---

Após a validação do desempenho da plataforma Arduino Mega, conclui-se que esta pode ser utilizada em projetos de UAV quadrotor para implementação de enxame de UAVs, que possuam os requisitos descritos neste trabalho.

E respondendo a questão proposta no começo deste trabalho sobre os dispositivos necessários a um projeto de UAV quadrotor que cumpra com os requisitos de controle de seus motores, percepção de invasores, percepção de obstáculos, localização geográfica, monitoramento de saúde e capacidade de comunicação, serão necessários: magnetômetro, giroscópio e acelerômetro para um controle efetivo dos motores; para percepção dos invasores pode-se utilizar uma câmera e algoritmos de visão computacional; para percepção de obstáculos pode-se utilizar sensores de proximidade como sonar; para localização geográfica utiliza-se o GPS; para monitoramento de saúde pode-se utilizar um circuito monitor de bateria;

Nota-se então que nem todos os dispositivos necessários para cumprir com os requisitos foram estudados neste trabalho. Os estudos sobre esses dispositivos serão realizados posteriormente.

## **7.1 Trabalhos futuros**

Realizar estudos sobre os outros dispositivos necessários a percepção de obstáculos e monitoramento de saúde, como monitoramento da quantidade atual de carga da bateria.

Realizar estudos e simulações de topologias e padrões de redes Ad Hoc para implementação da comunicação do enxame de UAVs.

Modificar os algoritmos do projeto (AeroQuad, 2011) para serem executados no microcontrolador NXP LPC1768 com o propósito de analisar a melhoria de desempenho que se obtém ao utilizar um microcontrolador mais rápido.

Embarcar algoritmos de visão computacional para reconhecimento de alvos para realização de monitoramento territorial.

Validar se o microcontrolador NXP LPC1768 consegue processar todos os algoritmos testados no experimento do capítulo 5, junto com os algoritmos de visão computacional.

---

Realizados experimentos de voo com o UAV quadrotor utilizando o controlador PID para controle dos motores e os dispositivos propostos neste trabalho.

Realizar experimentos de coordenação entre UAVs quadrotor utilizando o conceito de controle de enxames de UAVs implementados no simulador proposto por (Pinheiro, 2011).

Implementar e testar controladores para os motores baseados em redes neurais e lógica Fuzzy.

---

# Bibliografia

- A. P. (29 de 09 de 2011). *AeroQuad*. Acesso em 29 de Outubro de 2011, disponível em The Open Source Quadcopter: <http://aeroquad.com/>
- A. P. (10 de 09 de 2011). *ArduCopter*. Acesso em 29 de Outubro de 2011, disponível em Arduino-based autopilot for multitorotor craft, from quadcopters to traditional helis: <http://code.google.com/p/arducopter/wiki/ArduCopter>
- AeroQuad. (2011). *AeroQuad*. Acesso em 29 de Outubro de 2011, disponível em The Open Source Quadcopter: <http://aeroquad.com/>
- AeroQuad\_Sources. (06 de 12 de 2011). *AeroQuad*. Fonte: Google Code - AeroQuad: <http://code.google.com/p/aeroquad/>
- AeroQuad-Parts-List. (11 de Novembro de 2011). *AeroQuad*. Acesso em 14 de Novembro de 2011, disponível em AeroQuad Parts List: <http://aeroquad.com/showwiki.php?title=Parts+List>
- Alcatel-Lucent. (2009). Soluciones de comunicación Alcatel-Lucent Office para pequeñas y medianas empresas. *Manual de soluciones*.
- Arduino. (29 de Novembro de 2011). *Arduino Mega 2560*. Acesso em 29 de Outubro de 2011, disponível em Arduino Board Mega 2560: [arduino.cc](http://arduino.cc)
- ARM. (1 de 12 de 2011). *Cortex-M3 Processor*. Fonte: ARM: <http://www.arm.com/products/processors/cortex-m/cortex-m3.php>
- Atmel. (11 de Novembro de 2011). ATmega1280/2560 Datasheet. *ATmega640/1280/1281/2560/2561*.
- Bosch. (15 de Oct de 2009). BMP085 - Digital Pressure Sensor. *Datasheet BST-BMP085-DS000-05*.
- Camazine, S., Deneubourg, J., Franks, N., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2001). *Self-Organisation in Biological Systems*. New Jersey, USA: Princeton University Press.
- Costa, C. d. (2009). *Projeto de Circuitos Digitais com FPGA*. São Paulo: Érica Ltda.

---

Digikey. (11 de Novembro de 2011). *Digi-key Corporation*. Acesso em 11 de Novembro de 2011, disponível em Digikey: [www.digikey.com](http://www.digikey.com)

DIY-Drones. (01 de Dezembro de 2011). *DCM IMU Theory: First Draft*. Fonte: DIY-Drones: <http://diydrones.com/profiles/blogs/dcm-imu-theory-first-draft>

Editora Abril. (06 de Dezembro de 2011). *Como funciona o altímetro?* Fonte: Mundo Estranho - Editora Abril: <http://mundoestranho.abril.com.br/materia/como-funciona-o-altimetro>

Eletronic-Labs. (26 de 11 de 2011). *How to rebuild a Li-Ion battery pack*. Fonte: Eletronic-Labs: [http://www.electronics-lab.com/articles/Li\\_Ion\\_reconstruct/](http://www.electronics-lab.com/articles/Li_Ion_reconstruct/)

Everything2. (26 de 11 de 2011). *energy density*. Fonte: Everything2: <http://everything2.com/index.pl?node=energy%20density>

Fayzullin, M. (16 de Outubro de 2011). *Game Boy - Technical Information*. Acesso em 29 de Outubro de 2011, disponível em Marat Fayzullin Website: <http://fms.komkon.org/GameBoy/Tech/>

Halliday, D., Resnick, R., & Walker, J. (2009). *Fundamentos de Física 3: Eletromagnetismo* (7ª ed.). LTC.

Hoffmann, F. a. (2010). Attitude estimation and control of a quadrocopter. pp. 1072-1077.

Honeywell. (March de 2011). 3-Axis Digital Compass IC. *HMC5883L*.

HowStuffWorks. (01 de Dezembro de 2011). *How Electric Motors Work*. Fonte: HowStuffWorks: <http://electronics.howstuffworks.com/motor.htm/printable>

IEEE. (06 de Dezembro de 2011). *IEEE 802.15 WPAN™ Task Group 4 (TG4)*. Fonte: <http://www.ieee802.org/15/pub/TG4.html>

InvenSense. (2008). IDG-500. *Integrated Dual-Axis Gyro DS-IDG-0500B-00-01*.

InvenSense. (08 de July de 2009). ISZ-500. *Single-Axis Z-Gyro Product Specification PS-ISZ-0500B-00-03*.

InvenSense. (30 de March de 2010). ITG-3200. *Product Specification PS-ITG-3200A-00-01.4*.

- 
- Kennedy, J., & Eberhart, R. (November de 2005). Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference on* , pp. 1942 - 1948 vol.4 .
- Leptron. (27 de 11 de 2011). *SAFETY INFORMATION AND HANDLING PRECAUTIONS*. Fonte: Leptron - Custom Power Sources: <http://www.leptron.com/safety.do>
- Li, J., & Li, Y. (7-10 de August de 2011 ). Dynamic analysis and PID control for a quadrotor . *Mechatronics and Automation (ICMA), 2011 International Conference on*, pp. 573 - 578 .
- Malima. (06 de Dezembro de 2011). *O que é GPS? Como funciona?* Fonte: Malima: [http://www.malima.com.br/article\\_read.asp?id=237](http://www.malima.com.br/article_read.asp?id=237)
- MBED. (29 de Novembro de 2011). *MBED*. Acesso em 29 de Outubro de 2011, disponível em MBED -- Rapid Prototyping for Microcontroller: <http://mbed.org/>
- Mejias, L., Correa, J., Mondragon, I., & Campoy, P. (April de 2007). COLIBRI: A vision-Guided UAV for Surveillance and Visual Inspection. *Robotics and Automation, 2007 IEEE International Conference on* , pp. 2760-2761. Fonte: <http://www.vision4uav.com/>
- Melo, A. S. (30 de Junho de 2010). Dissertação de mestrado - UFES. *Implementação de um Quadrotor como Plataforma de Desenvolvimento para Algoritmos de Controle*. Vitória, Espírito Santo, Brasil.
- Microchip Technology. (2003). AN885 - DS00885A. *Brushless DC (BLDC) Motor Fundamentals*.
- Moore, A. (14 de Novembro de 2008). *Lithium Polymer (Lipo) Battery Guide*. Acesso em 14 de Novembro de 2011, disponível em Protalk.net - Engineers helping Engineers: <http://prototalk.net/forums/showthread.php?t=22>
- Morimoto, C. E. (20 de Fevereiro de 2007). *Entendendo os sistemas embarcados*. Acesso em 30 de Outubro de 2011, disponível em Hardware: <http://www.hardware.com.br/artigos/entendendo-sistemas-embarcados/>

- 
- Multipilot-32. (29 de 09 de 2011). *Multipilot 32*. Acesso em 11 de Novembro de 2011, disponível em Virtualrobotix: <http://www.virtualrobotix.com/page/multipilot32-1>
- Multipilot-8. (29 de 09 de 2011). *Multipilot 8 v1.0*. Acesso em 11 de Novembro de 2011, disponível em Virtualrobotix: <http://www.virtualrobotix.com/page/multipilot-8-v10>
- NXP-LPC17XX. (19 de Agosto de 2010). UM10360. *LPC17XX User manual*.
- NXP-LPCXpresso. (14 de Junho de 2011). LPCXpresso - User Guide. *Getting started with NXP LPCXpresso*.
- OIWTECH. (2010). Manual do Usuário. *AP OIW-2421APG*.
- Pinheiro, D. M. (Dezembro de 2011). Coordenação de Veículos Aéreos Não Tripulados Utilizando Inteligencia de Enxames. *Trabalho de Conclusão de Curso - Universidade de Pernambuco*. Recife, Pernambuco.
- Rufino, C. Z. (Junho de 2011). Desenvolvimento de um controlador embarcado com visão digital para o Robô-VD. *Trabalho de Conclusão de Curso - Universidade de Pernambuco*. Recife, Pernambuco.
- Sahin, E. (2005). Swarm Robotics: From Sources of Inspiration to Domains of Application. *Lecture Notes in Computer Science*, 10-20.
- Santos, I. F. (2001). *Dinâmica de Sistemas Mecânicos, Modelagens, Simulação*,. Makron Books.
- Santos, S. T. (Junho de 2007). REDES DE SENSORES SEM FIO EM MONITORAMENTO E CONTROLE. *Dissertação - UFRJ*. Rio de Janeiro, RJ.  
Fonte: Wikipedia: [http://en.wikipedia.org/wiki/IEEE\\_802.15.4](http://en.wikipedia.org/wiki/IEEE_802.15.4)
- SCHUNK, L. M., & LUPPI, A. (2001). *Microcontroladores AVR - Teoria e aplicações práticas*. Érica.
- Souza, D. J. (2000). *Desbravando o PIC*. São Paulo: Érica.
- Stallings, W. (2003). *Arquitetura e Organização de Computadores* (5ª ed.). São Paulo: Prentice Hall.
- STMicroelectronics. (July de 2009). LPR510AL. *Doc ID 15811 Rev 2*.

- 
- Strojny, B. a. (2009). Integration of conformal GPS and VHF/UHF communication antennas for small UAV applications. *Antennas and Propagation, 2009. EuCAP 2009. 3rd European Conference on*, pp. 2488-2492.
- Sullivan, J. M. (8-10 de June de 2005). Revolution or Evolution? The Rise of the UAVs. *Technology and Society, 2005. Weapons and Wires: Prevention and Safety in a Time of Fear. ISTAS 2005. Proceedings. 2005 International Symposium on*, pp. 94 - 101.
- Tanenbaum, A. S. (1999). *Organização Estrurada de Computadores (3ª ed.)*. Rio de Janeiro: LTC.
- Wikipedia. (26 de 11 de 2011). *Lithium-ion\_polymer\_battery*. Fonte: Wikipedia: [http://en.wikipedia.org/wiki/Lithium-ion\\_polymer\\_battery](http://en.wikipedia.org/wiki/Lithium-ion_polymer_battery)
- Wikipedia. (01 de 12 de 2011). *Microcontroller*. Fonte: Wikipedia: <http://en.wikipedia.org/wiki/Microcontroller>
- Wikipedia. (28 de 11 de 2011). *Unmanned\_aerial\_vehicle*. Fonte: Wikipedia: [http://en.wikipedia.org/wiki/Unmanned\\_aerial\\_vehicle](http://en.wikipedia.org/wiki/Unmanned_aerial_vehicle)
- Wikipedia. (28 de 11 de 2011). *Veículo\_aéreo\_não\_tripulado*. Fonte: Wikipedia: [http://pt.wikipedia.org/wiki/Veículo\\_aéreo\\_não\\_tripulado](http://pt.wikipedia.org/wiki/Veículo_aéreo_não_tripulado)

---

# **Apêndice A**

## **Implementação em C++ do algoritmo PSO**

PSO.h

```

/*
 *
 * Arquivo com implementação das classes Particula e PSO
 * para validação do desempenho da plataforma Arduino Mega.
 * Estas classes foram adicionadas ao projeto AeroQuad para
 * avaliação de tempo de processamento.
 *
 * Created on: 25/11/2011
 * Author: David Alain
 */

#ifndef PSO_H_
#define PSO_H_

#define NUM_VIZINHOS 2
#define VELOCIDADE_MAXIMA 50.0

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

class Particula{
public:
    float c1;
    float c2;
    float inercia;
    float pFitness;
    float gFitness;
    float posicao[2];
    float velocidade[2];

    float pBest[2];
    float gBest[2];

    inline float avalia(float alvo[2]){
        return sqrt( pow(posicao[0] - alvo[0],2) + pow(posicao[1] - alvo[1],2) );
    }

    void atualiza_velocidade(){

        //R1 e R2 são valores entre 0 e 1
        float r1 = (rand() % 101)/100.0;
        float r2 = (rand() % 101)/100.0;

        for (int i = 0; i < 2; i++) {
            velocidade[i] = inercia * velocidade[i] +
                c1 * r1 * (pBest[i] - posicao[i]) +
                c2 * r2 * (gBest[i] - posicao[i]);

            //Limita os valores em cada dimensão entre o mínimo e
            //o máximo permitido
            velocidade[i] = (velocidade[i] > VELOCIDADE_MAXIMA) ?
                VELOCIDADE_MAXIMA : velocidade[i];
            velocidade[i] = (velocidade[i] < -VELOCIDADE_MAXIMA) ?
                -VELOCIDADE_MAXIMA : velocidade[i];
        }
    }
}

```

```

void atualiza_posicao(){
    posicao[0] += velocidade[0];
    posicao[1] += velocidade[1];
}

void atualiza_pBest(){
    pBest[0] = posicao[0];
    pBest[1] = posicao[1];
}

void atualiza_gBest(float novo_gBest[2]){
    gBest[0] = novo_gBest[0];
    gBest[1] = novo_gBest[1];
}

void inicia(){
    c1 = 1;
    c2 = 1;
    inercia = 1;
    pFitness = 1000.00;
    posicao[0] = 0.1;
    posicao[1] = 0.1;
    velocidade[0] = 0.1;
    velocidade[1] = 0.1;

    pBest[0] = 0.6;
    pBest[1] = 0.6;
    gBest[0] = 0.6;
    gBest[1] = 0.6;
}

};

class PSO{
private:
    float alvo[2];
    Particula eu;
    Particula particulas[2];

public:

    void inicia(){
        for(int i = 0 ; i < NUM_VIZINHOS ; i++){
            particulas[i].inicia();
        }
        eu = particulas[0];
    }
}

```

```
void itera(){

    //Atualiza o alvo
    alvo[0] = (rand() % 101)/1.0;
    alvo[1] = (rand() % 101)/1.0;

    //Atualiza o pBest
    float valorFitness = eu.avalia(alvo);
    if(valorFitness < eu.pFitness){
        eu.atualiza_pBest();
    }
    eu.pFitness = valorFitness;

    //Atualiza o gBest
    for(int i = 0 ; i < NUM_VIZINHOS ; i++){
        particulas[i].atualiza_pBest();
        valorFitness = particulas[i].avalia(alvo);
        if(eu.gFitness > valorFitness){
            eu.gFitness = valorFitness;
            eu.gBest[0] = particulas[i].posicao[0];
            eu.gBest[1] = particulas[i].posicao[1];
        }
    }

    eu.atualiza_velocidade();
    eu.atualiza_posicao();

}

};

#endif /* PSO_H_ */
```