

# Universidade do Algarve

## Faculdade de Ciências e Tecnologia

**Aeromodelo telecomandado controlado remotamente através da Internet para  
caracterização do ambiente através de imagens e vídeo**

João Pedro Rosado Landeiro

Mestrado em Engenharia Eletrónica e Telecomunicações

2012



# Universidade do Algarve

## Faculdade de Ciências e Tecnologia

### **Aeromodelo telecomandado controlado remotamente através da Internet para caracterização do ambiente através de imagens e vídeo**

João Pedro Rosado Landeiro

Mestrado em Engenharia Eletrónica e Telecomunicações

Dissertação orientada por:

Professor Dr. Hélder Aniceto Amadeu de Sousa Daniel

Professor Dr. Luís Miguel S. C. Moura

**2012**



# Declaração

João Landeiro, estudante do mestrado em Engenharia Eletrónica e Telecomunicações na Universidade do Algarve, declara ser o autor deste documento.

**Estudante:**

---

(João Pedro Rosado Landeiro)

**Supervisor:**

---

(Professor Dr. Hélder Aniceto Amadeu de Sousa Daniel)

# Agradecimentos

Quero começar por agradecer aos meus orientadores, ao professor Dr. Hélder Aniceto Amadeu de Sousa Daniel pelo apoio prestado, e pelo tempo despendido durante toda a fase de desenvolvimento da dissertação.

Ao professor Dr. Luís Miguel S. C. Moura pelos seus comentários assertivos e apoio financeiro prestado.

Quero fazer um agradecimento especial ao Sr. António Quintas por todo o conhecimento e apoio prestado na construção e condução do aeromodelo.

Quero agradecer aos meus pais, por todo o esforço que fizeram ao longo dos anos para me manter a estudar, aos meus amigos e colegas pelo apoio prestado, em especial ao Ricardo Luz e ao Aldric Négrier.

Quero ainda agradecer ao Eng. Artur Neves pelo material disponibilizado, aos técnicos Fred Zabel e Celestino Martins pelo apoio técnico prestado, e aos serviços da oficina de mecânica da Universidade do Algarve.

# Resumo

O objetivo desta dissertação é controlar um aeromodelo (avião) telecomandado (AMT) por rádio frequência através da internet. Foi implementado e explorado o controlo do AMT utilizando duas infraestruturas distintas, a WLAN (*wireless local area network*) e a rede móvel de telecomunicações celulares ou WWAN (*wide wireless area network*), mantendo também o controlo clássico por rádio frequência ativo, visto que será útil como sistema de segurança em caso de perda de sinal de rede ou troca manual para o modo clássico. Desta forma no total existem três modos de controlo remoto: o clássico (por rádio frequência), por WLAN e por WWAN. No modo clássico o utilizador efetua o controlo através do telecomando com a tecnologia FHSS (*frequency-hopping spread spectrum*) na gama dos 2.4 GHz, que comunica com o recetor de rádio frequência com a mesma tecnologia, que se encontra no AMT. No modo WLAN o AMT encontra-se ligado à internet através de uma rede sem fios. Já no modo WWAN o AMT encontra-se ligado à internet através de uma rede de telecomunicações celulares. Em qualquer destes dois modos o utilizador pode estar ligado com um computador à internet por qualquer um dos métodos existentes atualmente (Ethernet, WLAN, Rede de telecomunicações celulares (WWAN)). Nestes casos o utilizador comanda o AMT através de um interface gráfico onde pode enviar comandos de atitude para o AMT usando um teclado ou um *joystick*, recebendo neste interface imagens provenientes da câmara instalada no AMT.

O computador de bordo do AMT recebe os comandos empacotados num protocolo específico e aplica-os a um controlador de servos que posiciona o leme de direção e o leme de profundidade, e define a aceleração do motor.

Esta dissertação tem como motivação o facto de os vários modos de controlo implementados permitirem uma experiência de voo à distância (fora da linha de vista), quer para um utilizador que simplesmente quer pilotar o AMT, quer para quem lhe quiser dar um uso mais orientado para a monitorização de espaços exteriores.

## Palavras-chave

Aeromodelo telecomandado, *wireless local area network*, rede de telecomunicações celulares, controlo remoto.

# Abstract

The goal of this dissertation is to control a radio frequency remote controlled model airplane (RCMA) using the internet. The control of the RCMA was implemented and explored using two different infrastructures, the WLAN (wireless local area network) and the mobile telecommunication cellular network, also known as WWAN (wide wireless area network), always keeping the radio frequency control activated, as it will be used as a safety system in case of low network signal strength, or as manual switch to classic mode. This way there are three remote control modes in total: the classic (by radio frequency), by WLAN and by WWAN. In the classic mode the user controls the RCMA using a FHSS (frequency-hopping spread spectrum) radio and receiver, that work in the 2.4 GHz frequency range.

In the WLAN mode the RCMA is connected to the internet through a wireless network. In the other hand, when in WWAN mode, the RCMA is connected to the internet through a mobile telecommunications cellular network. In each one of the previous modes the user can be connected to the internet by any of the actual means of access (Ethernet, WLAN, mobile telecommunications cellular network (WWAN)). In these cases the user controls the RCMA through a graphical user interface (GUI), where it can send attitude commands to the RCMA using either a keyboard or a joystick, and receive a video stream from the camera installed in the RCMA. The onboard computer of the RCMA receives the commands packed with a specific protocol, and transmits them to the servo controller that positions the rudder, the elevator, and sets the acceleration of the motor.

The motivation of this dissertation lies in the fact that the implemented control modes allow for an out of sight flight experience, either for a user that simply wants to pilot the RCMA, or for a user that wants monitor exterior spaces.

## Keywords

remote controlled model airplane, wireless local area network , mobile telecommunications cellular network, remote control.

# Conteúdo

## Índice

<b>1. Introdução .....</b>	<b>1</b>
1.1 Motivação .....	1
1.2 Estado de arte .....	1
1.2.1 Veículos telecomandados por rádio frequência .....	1
1.2.2 Redes de telecomunicação celular .....	3
1.3 Objetivos .....	6
<b>2. Tecnologias e Hardware.....</b>	<b>7</b>
2.1 Modelo de controlo de veículos telecomandados.....	7
2.1.1 Modelo de controlo clássico .....	7
2.1.2 Modelo de controlo por rede local sem fios.....	8
2.1.3 Modelo de controlo através de redes de telecomunicação celular .....	9
2.2 Componentes principais .....	10
2.2.1 Computador de bordo.....	10
2.2.2 Servomotores .....	11
2.2.3 Microcontrolador de servomotores .....	13
<b>3. Protótipo do aeromodelo.....</b>	<b>14</b>
3.1 Aeromodelo adotado .....	14
3.2 Construção do aeromodelo.....	15
3.3 Implementação dos componentes no aeromodelo.....	21
3.3.1 Instalação dos componentes comuns .....	21
3.3.2 Instalação dos componentes para o modo de controlo clássico .....	23
3.3.3 Instalação dos componentes para o modo de controlo remoto por WLAN e WWAN	24
<b>4. Análise do sinal de <i>pulse width modulation</i> .....</b>	<b>28</b>
4.1 Utilizando o recetor rádio.....	28

4.2	Utilizando o microcontrolador de servos Pololu.....	29
4.3	Utilizando multiplexagem dos dois modos referidos anteriormente.....	30
<b>5.</b>	<b>Tecnologias de <i>software</i> .....</b>	<b>32</b>
5.1	Modelo Cliente Servidor .....	32
5.2	<i>Live streaming media</i> .....	32
5.3	<i>Dynamic Domain Name Server</i> .....	33
<b>6.</b>	<b>Implementação do controlo do aeromodelo .....</b>	<b>34</b>
6.1	<i>Graphical User Interface</i> do cliente.....	34
6.2	Comunicação entre cliente e servidor .....	36
6.3	Descrição do funcionamento do cliente. ....	37
6.4	Descrição do funcionamento do servidor.....	38
6.5	<i>Video streaming</i> .....	40
6.5.1	Motion.....	40
<b>7.</b>	<b>Ensaio do aeromodelo.....</b>	<b>41</b>
7.1	Condução remota do aeromodelo.....	41
7.2	Condução remota utilizando uma WLAN.....	41
7.3	Condução remota utilizando uma rede de telecomunicação celular .....	41
7.4	Comparação entre os três modos de controlo.....	42
<b>8.</b>	<b>Conclusões e trabalho futuro .....</b>	<b>43</b>
	<b>Apêndice A: Configuração do <i>kernel</i> Linux .....</b>	<b>44</b>
	<b>Apêndice B: Configuração do Motion.....</b>	<b>46</b>
	<b>ANEXO A .....</b>	<b>48</b>
	<b>ANEXO B.....</b>	<b>60</b>
	<b>ANEXO C .....</b>	<b>61</b>
<b>9.</b>	<b>Bibliografia.....</b>	<b>62</b>

# Índice de Figuras

Figura 1.1: Aerosonde <i>Millionaire</i> .....	2
Figura 1.2: Rede de telecomunicação celular típica. ....	3
Figura 2.1: Modelo de controlo clássico. ....	7
Figura 2.2: Modelo de controlo por rede local sem fios. ....	8
Figura 2.3: Modelo de controlo através de redes de telecomunicações celulares.....	9
Figura 2.4: SBC Snowball (12).....	10
Figura 2.5: Servomotor típico. ....	11
Figura 2.6: Funcionamento de um servomotor. ....	11
Figura 2.7: Modelação por largura de pulso. ....	12
Figura 2.8: Pololu Micro Maestro.....	13
Figura 3.1: RQ-11 Raven (à esquerda), e Skyartec Cessna 182 (à direita).....	14
Figura 3.2: Primeiro esboço do aeromodelo. ....	16
Figura 3.3: Construção dos lemes e reforço das uniões dos lemes e das asas. ....	16
Figura 3.4: Melhoramento da fuselagem, cortes e reforço da fuselagem e do leme de direção. ....	17
Figura 3.5: Reforço em contraplacado no leme de profundidade. ....	17
Figura 3.6: Visualização do espaço interior da fuselagem.....	18
Figura 3.7: Preparação da colagem dos lemes. ....	18
Figura 3.8: Colagem dos lemes, e fixação do trem de aterragem. ....	19
Figura 3.9: Fixação do suporte do motor, do motor e da hélice. ....	19
Figura 3.10: Exterior do aeromodelo completo. ....	20
Figura 3.11: Suporte para a bateria <i>Li-Po</i> e para a bateria de 4.8 V que alimenta o HUB USB. ....	22
Figura 3.12: Suporte para o recetor rádio, controlador de servos, circuito do <i>multiplexer</i> , e <i>HUB USB</i> . ....	22
Figura 3.13: Instalação dos suportes no interior do AMT. ....	22
Figura 3.14: Suporte para fixação do SBC Snowball. ....	22
Figura 3.15: Recetor rádio de 6 canais.....	23
Figura 3.16: Fixação do recetor rádio já com os servomotores e o ESC ligados.....	23
Figura 3.17: Circuito de multiplexagem utilizado (13).....	25
Figura 3.18: Implementação do circuito de multiplexagem. ....	26
Figura 3.19: Sistema de ligações completo.....	27
Figura 4.1: Pulso mínimo (esquerda), Pulso máximo (direita) (Escala: X- 1 ms, Y- 2 volt). ....	28

Figura 4.2: Pulso médio (esquerda) (Escala: X- 1 ms, Y- 2 volt), Período de repetição de pulsos (direita) (Escala: X- 5 ms, Y- 2 volt). .....	28
Figura 4.3: Pulso mínimo (esquerda), Pulso máximo (direita) (Escala: X- 1 ms, Y- 2 volt). .....	29
Figura 4.4: Pulso médio (esquerda) (Escala: X- 1 ms, Y- 2 volt), Período de repetição de pulsos (direita) (Escala: X- 5 ms, Y- 2 volt). .....	29
Figura 4.5: Pulso mínimo (esquerda), Pulso máximo (direita) (Escala: X- 1 ms, Y- 2 volt). .....	30
Figura 4.6: Pulso médio (esquerda) (Escala: X- 1 ms, Y- 2 volt), Período de repetição de pulsos (direita) (Escala: X- 5 ms, Y- 2 volt). .....	30
Figura 4.7: Pulso mínimo (esquerda), Pulso máximo (direita) (Escala: X- 1 ms, Y- 2 volt). .....	31
Figura 4.8: Pulso médio (esquerda) (Escala: X- 1 ms, Y- 2 volt), Período de repetição de pulsos (direita) (Escala: X- 5 ms, Y- 2 volt). .....	31
Figura 6.1: GUI com comunicação por WLAN ou WWAN .....	35
Figura 6.2: GUI com comunicação através do modo clássico .....	35
Figura 6.3: Diagrama das tarefas executadas no cliente. ....	38
Figura 6.4: Diagrama do servidor. ....	38
Figura 6.5: Diagrama da tarefa principal. ....	39

# Índice de Tabelas

Tabela 3.1: Componentes para a construção do aeromodelo.....	15
Tabela 3.2: Componentes para o controlo remoto por <i>WLAN e WWAN</i> .....	24
Tabela 3.3: Tabela de verdade do <i>multiplexer</i> (13). ....	25
Tabela 6.1: Objetos enviados. ....	36
Tabela 7.1: Vantagens e desvantagens dos modos de controlo. ....	42

# Lista de acrónimos

8-PSK	<i>Eight-Phase Shift Keying</i>
AMPS	<i>Advanced Mobile Phone System</i>
AMT	<i>Aeromodelo Telecomandado</i>
API	<i>Application Programming Interface</i>
ARF	<i>Almost Ready To Fly</i>
ARM	<i>Advanced RISC Machine</i>
BEC	<i>Battery Elimination Circuit</i>
CDMA	<i>Code Division Multiple Access</i>
CDMA 1xEVDO	<i>Code Division Multiple Access Evolution Data Only</i>
DDNS	<i>Dynamic Domain Name Server</i>
DDR-SDRAM	<i>Double Data Rate-Synchronous Dynamic Random Access Memory</i>
DMA	<i>Direct Memory Access</i>
EDGE	<i>Enhanced Data Rates for GSM Evolution</i>
e-MMC	<i>Embedded Multimedia Card</i>
ESC	<i>Electronic Speed Controller</i>
FDMA	<i>Frequency Division Multiple Access</i>
FHSS	<i>Frequency-Hopping Spread Spectrum</i>
GPRS	<i>General Packet Radio Service</i>
GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile Communication</i>
GUI	<i>Graphical User Interface</i>
HSCSD	<i>High Speed Circuit Switched Data</i>
HSPA	<i>High Speed Packet Access</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IMT-Advanced	<i>International Mobile Telecommunications Advanced</i>
IP	<i>Internet Protocol</i>
ISP	<i>Internet Service Provider</i>
Li-ion	<i>Lithium-ion</i>
Li-Po	<i>Lithium-ion Polymer</i>
microSD	<i>Micro Secure Digital</i>
MSC	<i>Mobile Switching Center</i>
OTG	<i>On The Go</i>

PID	<i>Proportional Integral Derivative</i>
PSTN	<i>Public Switched Telephone Network</i>
PWM	<i>Pulse Width Modulation</i>
RCMA	<i>Remote Controlled Model Airplane</i>
RTF	<i>Ready To Fly</i>
RTSP	<i>Real-Time Streaming Protocol</i>
SBC	<i>Single Board Computer</i>
TCP	<i>Transmission Control Protocol</i>
TDMA	<i>Time Division Multiple Acces</i>
UAV	<i>Unmanned Aerial Vehicle</i>
UCAV	<i>Unmanned Combat Aerial Vehicle</i>
UDP	<i>User Datagram Protocol</i>
UMTS	<i>Universal Mobile Telecommunications System</i>
USB	<i>Universal Serial Bus</i>
WCDMA	<i>Wideband Code Division Multiple Access</i>
WLAN	<i>Wireless Local Area Network</i>
WWAN	<i>Wide Wireless Area Network</i>

# 1. Introdução

## 1.1 Motivação

Os veículos telecomandados por rádio frequência comercializados atualmente têm limitações físicas impostas pelas tecnologias de comunicação existentes e pelo meio de comunicação. Uma das limitações é a distância a que o utilizador efetua o controlo do veículo (que para rádios FHSS de 2.4 GHz é cerca de 1.2 Km). Outra limitação que é inerente à citada anteriormente é o facto de o controlo do veículo se efetuar em linha de vista o que limita muito o raio de ação.

Esta dissertação foca-se em remover estas limitações de controlo em veículos telecomandados por rádio frequência, adicionando dois modos de controlo remoto ao modo clássico.

## 1.2 Estado de arte

### 1.2.1 Veículos telecomandados por rádio frequência

O aparecimento de veículos telecomandados por rádio frequência remonta ao século XIX, quando em 1898 Nikola Tesla patenteou o primeiro barco telecomandado por rádio frequência (1) e apresentou-o na Exibição Elétrica em *Madison Square Garden*. Daí em diante foram vários os tipos de veículos telecomandados por rádio frequência a serem implementados, tais como aviões, carros, tanques, helicópteros, tendo a maior parte deles sido construídos durante a segunda guerra mundial.

O aparecimento das primeiras aeronaves remotamente controladas por rádio frequência, também conhecidos como UAVs (*unmanned aerial vehicles*), ou UCAVs (*unmanned combat air vehicles*) quando utilizados para fins de combate militar, deu-se em 1916 quando Archibald Montgomery Low lançou o "*Aerial Target*", um protótipo que ainda chegou a fazer alguns testes de voo, mas o projeto acabou por ser abandonado depois da primeira guerra mundial. No período entre a primeira guerra mundial e a segunda guerra mundial, foram desenvolvidos diversos modelos controlados remotamente, mas só depois da segunda guerra mundial é que foram construídos modelos com motor a jacto, como o *Teledyne Ryan Firebee I* em 1951 ou o *Model 1001* feito para a marinha dos Estados Unidos em 1955 pela companhia *Beechcraft*. Contudo estes modelos ainda não eram automatizados, isto é, precisavam de um piloto que os controlava remotamente, facto que só se veio a alterar nos modelos construídos durante e após a guerra do Vietname (2).

A automação de UAV's desenvolveu-se ao longo dos anos, mas até umas décadas atrás, a decolagem e a manobra de aterragem ainda eram efetuadas por pilotos. Em 1997 deu-se um passo muito importante na automação completa de UAV's, quando a Aerosonde *Millionaire* (Figura 1.1) voou em piloto automático desde a decolagem até à aterragem (3).



Figura 1.1: Aerosonde *Millionaire*.

A evolução de aeromodelos telecomandados comercializados ainda não está completamente automatizada, mas já existem alguns componentes, como giroscópio, acelerómetro, GPS (*global positioning system*), entre outros, que permitem a implementação de sistemas de controlo autónomo. O campo onde os aeromodelos mais evoluíram, foi na capacidade de tornar o sistema de telecomunicação muito tolerante a interferências. Através da implementação de múltiplos sistemas de *spread spectrum*, tais como o FHSS, é possível a utilização de um grande número de aeromodelos no mesmo espaço sem que exista interferência. Outro dos progressos foi na área de cobertura dos sistemas de telecomunicação, onde os sistemas mais sofisticados de radiotransmissores e recetores já conseguem cobrir uma área de mais de quatro Km.

## 1.2.2 Redes de telecomunicação celular

O conceito de redes de telecomunicação celular foi inventado no início da década de 1970 pela AT&T Bell Labs (4), mas o termo só começou a ser mais conhecido após a revolucionária invenção do primeiro telefone celular comercializado, por Martin Couper a 3 de abril de 1973, enquanto chefe executivo do departamento de telefones para carros da Motorola (5). Previamente a esta década as comunicações sem fios eram muito difíceis de estabelecer, visto que havia apenas uma torre transmissora central de elevada potência que disponibilizava cerca de cinco canais para uma zona metropolitana, e cerca de uma a oito torres recetoras, que faziam o retorno das chamadas, permitindo apenas três chamadas simultâneas (6).

As redes de telecomunicação celular são divididas em áreas de cobertura do serviço, que por sua vez são divididas em áreas hexagonais mais pequenas, que são designadas como células. Cada célula é servida por uma estação base (*base station*), que é fixa. Esta é capaz de comunicar com estações móveis tais como telemóveis, usando o seu transceptor.

A estação base está conectada ao centro de comutação móvel (MSC, *mobile switching center*), que está por sua vez conectado à rede pública de comutação telefónica (PSTN, *public switched telephone network*). A Figura 1.2 ilustra uma rede de telecomunicação celular típica, em que os símbolos dentro dos hexágonos são estações base.

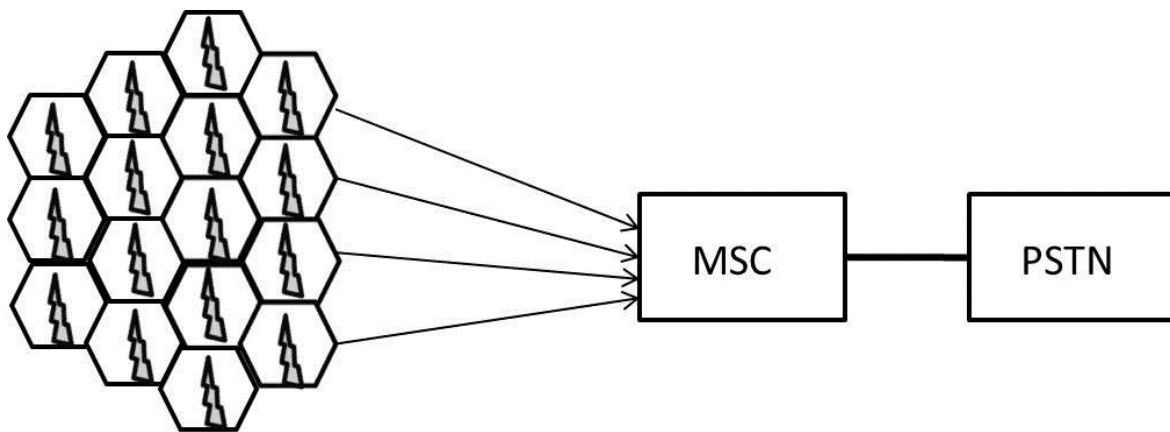


Figura 1.2: Rede de telecomunicação celular típica.

O espectro de frequências alocado para comunicações sem fios é muito limitado, então as redes de telecomunicação celular começaram a reutilizar frequências. A cada célula são atribuídos um número de canais de frequências. Para que não exista interferência, os canais atribuídos a uma célula não podem ser utilizados pelas suas células vizinhas, isto é, têm de estar suficientemente afastadas para que a interferência entre elas seja tolerável. Ao reduzir o tamanho das células, a rede

de telecomunicação celular aumenta o número de estações base, e portanto a sua capacidade, que se reflete na capacidade de servir mais utilizadores (7).

A evolução das tecnologias inerentes às redes de telecomunicação celular tem sido permanente. Desde o início das redes de telecomunicação celular, já contamos quatro gerações de tecnologias, conhecidas como: 1G, 2G, 3G, e 4G.

A geração 1G operava com sinais analógicos e utilizava comutação de circuitos. O primeiro sistema a ser implementado em larga escala foi o AMPS (*advanced mobile phone system*) (6), que utilizava FDMA (*frequency division multiple access*) para dividir o espectro de frequências em canais, e operava na gama dos 800 MHz. Este sistema era apenas para voz, e tinha muitas desvantagens a comparar com os padrões de hoje em dia, as conversações eram muito suscetíveis ao ruído, não eram encriptadas, e eram muito fácil intercetar a chamada. À medida que o número de utilizadores crescia, as redes precisavam de um aumento de capacidade, então foram inventados os sistemas digitais. Esta segunda geração de sistemas (2G), consistia entre outros, o sistema europeu GSM (*global system for mobile communication*), e o norte-americano CDMA (*code division multiple access*), que ficou conhecido como cdmaOne.

O sistema GSM era comutado por circuitos e utilizava TDMA (*time division multiple access*) para acomodar em cada banda portadora com 200 KHz de largura, oito *slots* portadores, cada um correspondente a um circuito. Cada banda portadora consegue suportar até oito comunicações simultâneas. Cada célula tipicamente suporta múltiplas bandas portadoras. As radiofrequências usadas nas bandas portadoras podem ser reutilizadas entre células, desde que os transmissores rádio que usam a mesma frequência não estejam em células adjacentes (4).

O sistema cdmaOne utiliza uma tecnologia com a designação de *spread spectrum*, na qual o espectro de radiofrequências é dividido em portadoras com 1.23 MHz de largura. A cada canal de voz é atribuído um código único dentro da portadora, e o sinal de voz espalha-se a uma taxa de transmissão de 1.23 Mbit/s. O código único atribuído, é utilizado para descodificar o sinal transmitido no lado do recetor, e permite a reutilização de frequências em todas as células, tendo assim uma maior capacidade do que o sistema GSM (4).

A partir da geração 2G surgiram várias evoluções da tecnologia GSM já existente, como a HSCSD (*high speed circuit switched data*), a GPRS (*general packet radio service*), e a EDGE (*enhanced data rates for GSM evolution*). Estas evoluções tiveram a designação de 2.5G e 2.75G (no caso da EDGE), pois não eram uma tecnologia nova. Estas tecnologias surgiram na mesma ordem cronológica que estão enunciadas no texto. A HSCSD foi implementada diretamente nas redes GSM e consistia em reservar os mesmos *slots* que antes eram para sinais de voz, mas desta vez alocavam-nos para transferir dados. Tinha a desvantagem de os *slots* ficarem ocupados, não permitindo que outros utilizadores utilizassem a comunicação ao mesmo tempo da chamada de

dados (4). Como ainda não havia diferenciação de tipo de tráfego, o sistema TDMA do GSM taxava as chamadas de dados como se fossem chamadas de voz, o que o tornava num serviço muito dispendioso.

GPRS foi a tecnologia que introduziu comutação de pacotes nas redes de telecomunicação celular, permitindo assim reservar os recursos da rede apenas quando são necessários, flexibilizando a rede para permitir taxas de transferência mais altas quando os recursos estão disponíveis. Na tecnologia GPRS a taxa de transferência teórica máxima é de 160 kbit/s, mas na prática tem valores um pouco acima dos 40 kbit/s (8). A taxação de tarifas aqui já era feita através de quantidades de tráfego e não de tempo como na TDMA.

A tecnologia EDGE vem implementar uma nova técnica de transmissão 8-PSK (*eight-phase shift keying*) que triplica a velocidade de transferência do GPRS. É por vezes considerada uma rede de terceira geração, já que consegue ter velocidade suficiente para suportar uma vasta gama de serviços de dados, como o acesso a vídeos, websites, e *email*. Na realidade é caracterizada como uma tecnologia 2.75G, pois oferece uma velocidade de transferência de dados inferior a uma rede de telecomunicação celular 3G nativa (9).

Na geração 3G também existem vários sistemas adotados, sendo os dois com maior expressão mundial, o UMTS (*universal mobile telecommunications system*), e o CDMA 1xEVDO (*code division multiple access evolution data only*). Como o sistema implementado em Portugal é o UMTS, este vai ser o tópico desenvolvido. O UMTS foi desenvolvido através do WCDMA (*wideband code division multiple access*) para fornecer velocidades de acesso até 2 Mbit/s, permitindo o acesso a aplicações mais exigentes do ponto de vista da capacidade da rede, como o *streaming* multimédia. Tal como na geração 2G, também aqui existiram evoluções, sendo a mais relevante a introdução do padrão HSPA (*high speed packet access*) que apresenta velocidades de *download* de 14.4 Mbit/s, e 2 Mbit/s de *upload*. Além do aumento das taxas de transferência de dados e da capacidade da rede, o HSPA também apresenta tempos de latência muito inferiores aos sistemas anteriores, tornando esta tecnologia já atrativa para aplicações de controlo remoto de veículos, como é este o caso (10).

As redes de quarta geração (4G), são ainda uma novidade no nosso mercado e estão longe de atingir os valores máximos de taxas de transferência especificados pelo IMT-Advanced (*international mobile telecommunications advanced*), que fixam um máximo de 100 Mbit/s para utilizadores com mobilidade elevada (carros, comboios), e de 1 Gbit/s para utilizadores com baixa mobilidade (pedestres). No entanto as redes 4G já permitem que os utilizadores façam uso de aplicações exigentes, como a televisão móvel em alta definição, a videoconferência, e a televisão 3D (11). Esta é a solução mais atrativa em termos de performance, mas o preço a pagar pelos

equipamentos e tarifários ainda não está ao alcance de todos, por isso esta geração ainda foi posta de parte na infraestrutura do AMT utilizado nesta dissertação.

### 1.3 Objetivos

Um dos objetivos desta dissertação é construir uma aplicação que implementa os dois modos de controlo remoto, através de uma WLAN (*wireless local area network*) e através de uma rede de telecomunicação celular ou WWAN (*wide wireless area network*), bem como multiplexar o modo clássico com um dos dois modos remotos acima referidos, usando uma *flag* (sinal lógico com dois estados *high* ou *low*) de controlo para garantir que o circuito de multiplexagem faz a comutação para o modo clássico em caso de falha de rede. Um modo manual de comutação também foi implementado em caso de as condições atmosféricas não estarem muito favoráveis, dado que o modo clássico não tem praticamente atraso de propagação entre o emissor e o recetor.

O outro objetivo é garantir que imagens são armazenadas no computador de bordo e o vídeo é transmitido praticamente em tempo real, garantindo assim que é possível efetuar o controlo do AMT fora de linha de vista, expandindo assim as aplicações possíveis para o AMT.

## 2. Tecnologias e Hardware

### 2.1 Modelos de comunicação de veículos telecomandados

#### 2.1.1 Modelo de controlo clássico

O modelo de controlo clássico é implementado com três componentes principais, o emissor, o meio de transmissão e o recetor (Figura 2.1). O sinal propaga-se através de ondas rádio, que são um tipo de ondas eletromagnéticas. Existem várias modelações e técnicas de transmissão que podem ser utilizadas, mas para o propósito desta dissertação a modelação utilizada foi a PWM (*pulse width modulation*), que consiste na transmissão periódica de um pulso de largura variada e amplitude constante em cada canal do emissor e do recetor. Por exemplo se quisermos acelerar, no canal respetivo vamos transmitir um pulso com uma largura maior que o pulso anterior, se quisermos desacelerar transmitimos um pulso com largura inferior ao anterior. Acontece o mesmo para os outros comandos do AMT, se quisermos mudar de direção para cima ou para a direita, o pulso aumenta de largura, se quisermos mudar de direção para baixo ou para a esquerda, o pulso diminui de largura. Para que não existam interferências e seja feito um bom uso da largura de banda disponível, o emissor e o recetor implementam a tecnologia FHSS, que consiste numa portadora que troca rapidamente de canais de frequência através de uma sequência pseudoaleatória conhecida tanto pelo emissor como pelo recetor, tornando assim muito difícil a interceção do sinal, a não ser que se conheça a sequência.

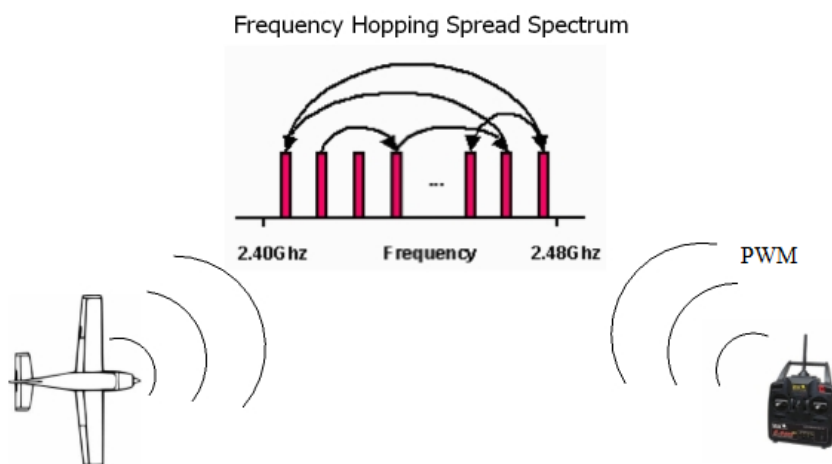


Figura 2.1: Modelo de controlo clássico.

## 2.1.2 Modelo de controlo por rede local sem fios

Este modelo consiste numa comunicação entre cliente e servidor em que ambas as máquinas têm um interface de rede que permite a ligação à rede local sem fios (Figura 2.2). O servidor disponibiliza pelo menos um serviço ao qual o cliente pode aceder e controlar o veículo, neste caso um AMT. O AMT tem de ser adaptado com alguns componentes extra para garantir o desempenho desejado, um SBC (*single board computer*) com interface de rede sem fios, para executar o servidor, e também interface USB para ligar componentes de controlo do AMT como o microcontrolador de servos por exemplo.

A grande vantagem deste modo de controlo é que o AMT pode ser controlado por qualquer utilizador ligado à WLAN ou à internet se o computador de bordo do servidor também estiver, garantindo um sistema de controlo sem custos para o utilizador.



Figura 2.2: Modelo de controlo por rede local sem fios.

### 2.1.3 Modelo de controlo através de redes de telecomunicação celular

O modelo de controlo através de redes de telecomunicação celular também efetua a comunicação entre cliente e servidor, sendo a principal diferença o meio de acesso à internet no servidor. O servidor instalado no AMT liga-se à internet através de um interface de acesso a redes de telecomunicação celular, tal como um modem 3G com tecnologia HSPA.

O cliente pode ligar-se diretamente à internet acedendo a uma WLAN com acesso à internet, ou através de uma rede de telecomunicação celular, acedendo ao servidor. As duas opções são válidas, aliás qualquer meio de acesso à internet o é. Através deste modelo de controlo o AMT pode ser controlado por qualquer computador com acesso à internet, e como este tipo de rede cobre áreas muito extensas, é o ideal para o propósito desta dissertação. O modelo está representado na Figura 2.3.

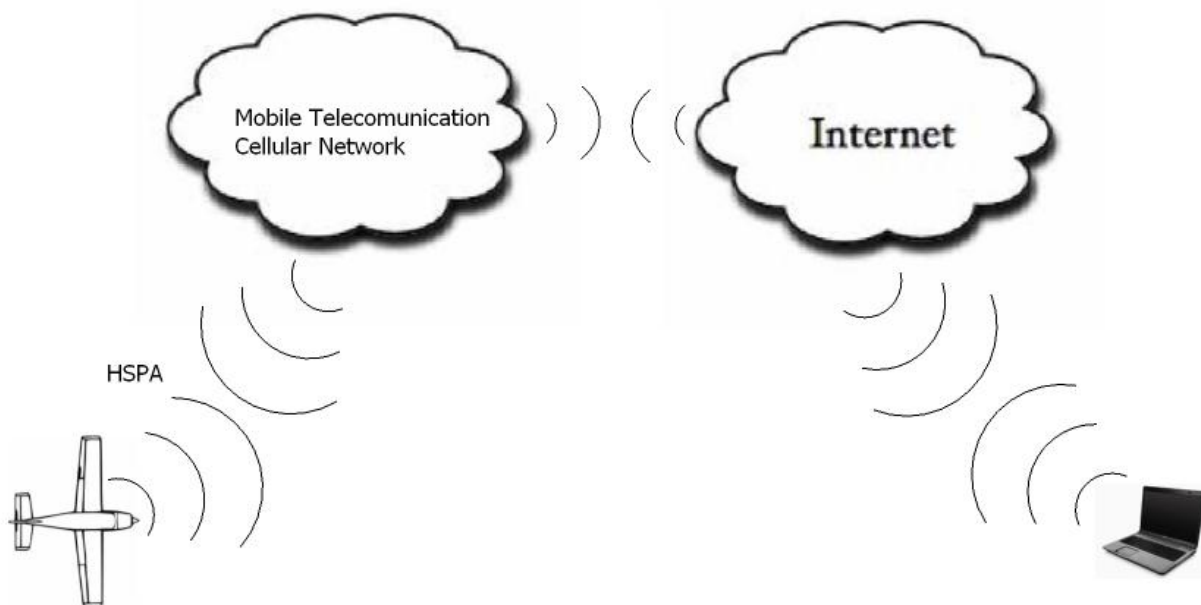


Figura 2.3: Modelo de controlo através de redes de telecomunicações celulares.

## 2.2 Componentes principais

### 2.2.1 Computador de bordo

O computador de bordo utilizado, foi o SKY-S9500-ULP-C12 (Conhecido como Snowball) que pelas suas especificações nos dá garantias de desempenho, e de resistência ao choque e vibrações, tudo graças à sua implementação numa só placa com dimensões reduzidas. Este pequeno computador é baseado na arquitetura ARM (*Advanced RISC Machine*) e tem entre outras especificações um processador ST-Ericsson Nova A9500 a 1GHz, 1Gb DDR-SDRAM, um disco eMMC de 4 Gb, um slot microSD que permite maior capacidade de armazenamento, uma bateria *Li-ion* com elevada autonomia (várias horas), interface de redes sem fios, e ainda um interface USB 2.0 OTG (*on the go*) que suporta uma taxa de transferência de 480 Mbit/s, permitindo assim ligar o HUB USB que faz o interface com todos os periféricos necessários para a implementação do tópico principal desta dissertação. O Snowball tem ainda a vantagem de poder correr várias distribuições do sistema operativo Linux e Android.

O sistema operativo utilizado foi o Linaro Ubuntu 12.03, tendo em vista a compatibilidade do *hardware* utilizado. O seu *kernel* teve de ser recompilado de modo a suportar a carga da bateria Li-ion, e a reconhecer dispositivos USB. A configuração encontra-se no Apêndice A.

Abaixo encontra-se a Figura 2.4 que revela o aspeto e indica alguns dos componentes mais importantes.

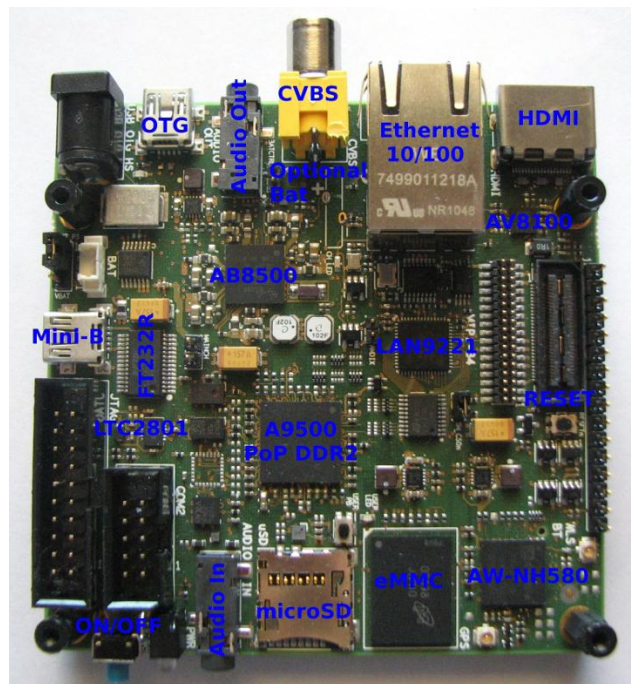


Figura 2.4: SBC Snowball (12).

## 2.2.2 Servomotores

O funcionamento de um servomotor típico como o da Figura 2.5 é feito em malha fechada, ou seja tem realimentação negativa para que seja efetuada a comparação entre a posição atual e a posição seguinte. Esta posição é determinada pela largura do pulso de controlo que é um sinal PWM. Esta largura de pulso é traduzida em tensão por um conversor e comparada com a tensão da posição atual. Conforme a variação de tensão na saída do comparador, assim o eixo do motor se moverá para a esquerda ou para a direita. Este funcionamento é bem perceptível no diagrama de blocos da Figura 2.6.



Figura 2.5: Servomotor típico.

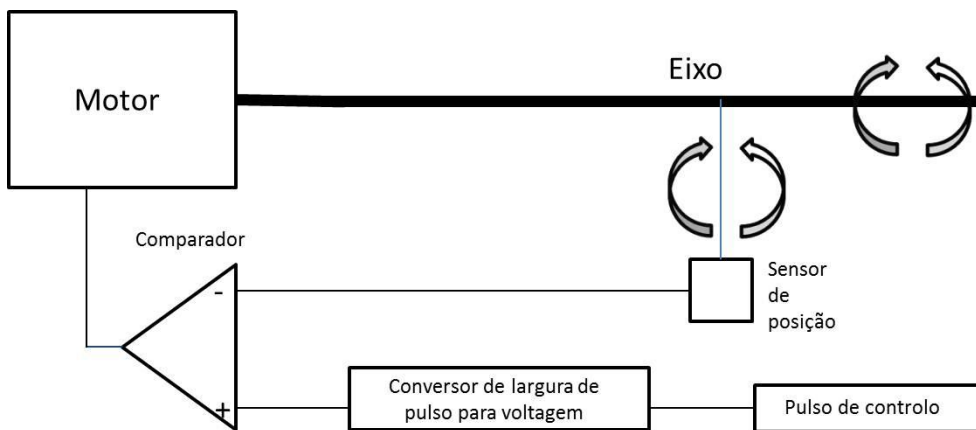


Figura 2.6: Funcionamento de um servomotor.

Os servomotores tipicamente têm uma liberdade de  $180^\circ$ , na Figura 2.7 podemos observar como a modelação por largura de pulso se traduz em movimento angular nos servomotores: viram para a esquerda caso a largura de pulso da posição seguinte seja menor do que a posição atual, e para a direita se a largura de pulso da posição seguinte for maior do que a posição atual.

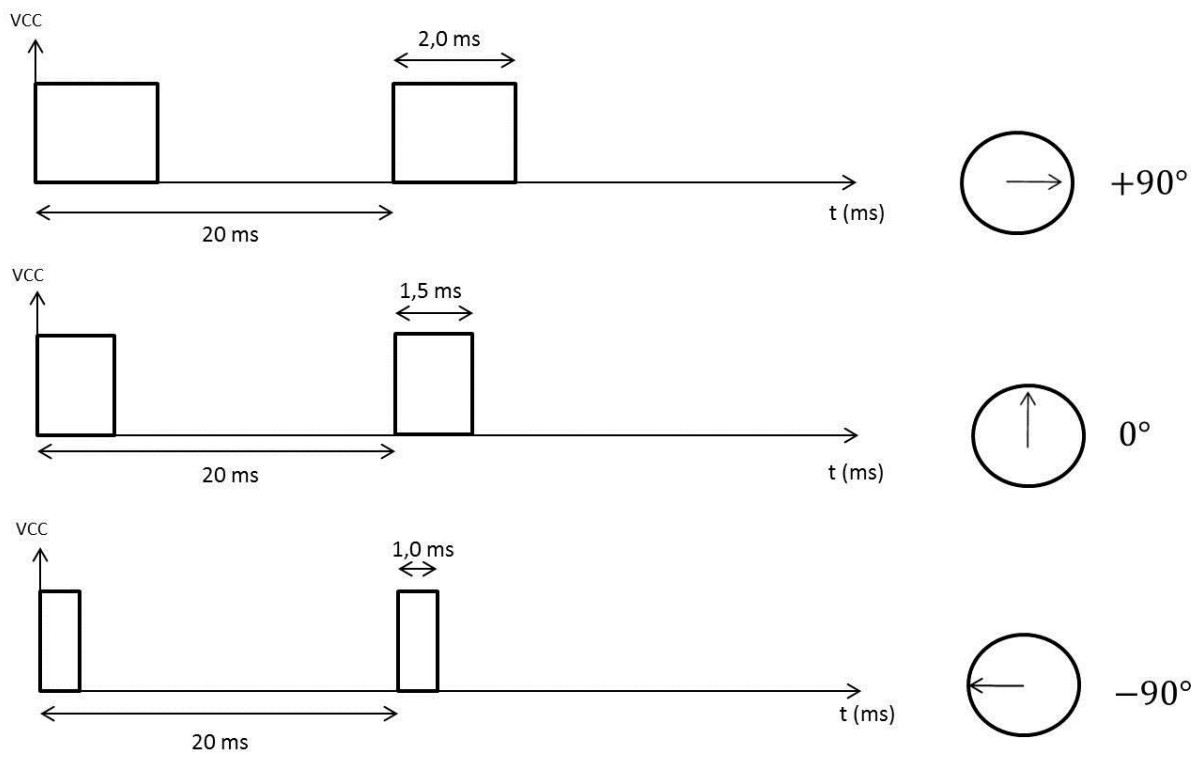


Figura 2.7: Modelação por largura de pulso.

### 2.2.3 Microcontrolador de servomotores

O Pololu Micro Maestro é um microcontrolador de servos de 6 canais que tem como funções interpretar os comandos enviados pelo servidor através da API (*application programming interface*) em Java RXTXComm, e transformar os dados em pulsos de largura variada, caso os canais estejam configurados como *servos*, ou em tensão, (zero ou cinco volt) caso o canal esteja configurado como *output*.

Este pequeno microcontrolador requer apenas uma alimentação de cinco volt para que os seus canais funcionem como descrito no parágrafo acima.

Abaixo encontra-se a Figura 2.8 que mostra o microcontrolador, suas ligações e indicadores.

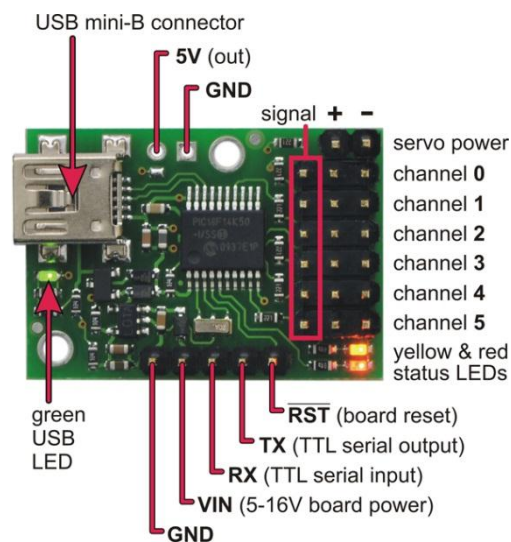


Figura 2.8: Pololu Micro Maestro.

# 3. Protótipo do aeromodelo

## 3.1 Aeromodelo adotado

O aeromodelo foi construído propositadamente para este projeto, com apoio no conhecimento de construção de aeromodelos do Sr. António Quintas, tendo em conta quer o espaço requerido para acomodar todos os componentes de forma eficiente, quer aspetos como a aerodinâmica e tempo de voo.

O resultado desta combinação de fatores foi um aeromodelo híbrido entre um planador e um *trainer* com o perfil de asa curvo na parte de cima e quase reto na parte de baixo, conseguindo assim ter uma boa sustentação e estabilidade e ainda uma capacidade de planar satisfatória. Olhando para o mercado podemos encontrar alguns aeromodelos em que nos inspiramos. Um deles foi o RQ-11 Raven, que nos deu a perceção de como construir um perfil de asa em diedro (ângulo para cima que um aeromodelo de asa fixa faz entre a asa na horizontal e a parte da asa que fica inclinada), de forma a que o comando direcional do aeromodelo seja efetuado sem a necessidade de ailerons. Já a fuselagem do aeromodelo foi mais inspirada num modelo do tipo Cessna, como o Skyartec Cessna 182, de forma a ter espaço para acomodar os componentes necessários.



Figura 3.1: RQ-11 Raven (à esquerda), e Skyartec Cessna 182 (à direita).

## 3.2 Construção do aeromodelo

Tabela 3.1: Componentes para a construção do aeromodelo.

Material	Imagem	Material	Imagem
Placas em poliestireno extrudido ( <i>wallmate</i> )		Cola para madeira	
Espuma de poliuretano		Cola Araldite	
Tubos em alumínio		Vareta de aço	
Contraplacado		Xizato	
Balsa		Lixa	
Cartolina		<i>Clevises</i> e varetas de alumínio	
Rodas		Dobradiças de plástico	

A construção do aeromodelo foi efetuada reutilizando grande parte de materiais já usados, como foi o caso das colas, das placas de poliestireno extrudido e de contraplacado, conseguindo assim ter uma excelente relação qualidade/preço.

No início começou-se por construir o corpo e as asas do aeromodelo, utilizando apenas o x-ato, a lixa e a espuma de poliuretano. Logo de seguida foram feitos o suporte para os lemes de direção e de profundidade de modo a criar o esboço da Figura 3.2.



Figura 3.2: Primeiro esboço do aeromodelo.

Como a construção do aeromodelo teve de ser faseada, devido à secagem das colas optou-se por fazer várias partes ao mesmo tempo. Enquanto umas secavam faziam-se outras partes do trabalho. O próximo passo foi cortar os lemes, reforçar as uniões com contraplacado e inserir as dobradiças, bem como reforçar as asas com contraplacado e balsa, como pode ser visto na Figura 3.3.



Figura 3.3: Construção dos lemes e reforço das uniões dos lemes e das asas.

Nesta fase da construção a fuselagem foi lixada de modo a suavizar os contornos, e também foram efetuados os cortes para ter acesso aos componentes que vão ser instalados no futuro. Ao mesmo tempo que foram adicionados reforços em contraplacado na fuselagem para o encaixe com as asas, bem como no leme de direção para o apoio que vai permitir o controlo através de um servomotor. Como se pode ver na Figura 3.4.



Figura 3.4: Melhoramento da fuselagem, cortes e reforço da fuselagem e do leme de direção.

Ao mesmo tempo, também o leme de profundidade era reforçado com um pouco de contraplacado para a interação com o outro servomotor, como se vê na Figura 3.5.



Figura 3.5: Reforço em contraplacado no leme de profundidade.

Na Figura 3.6 pode-se ver como o interior da fuselagem é espaçoso, permitindo a futura acomodação dos componentes.

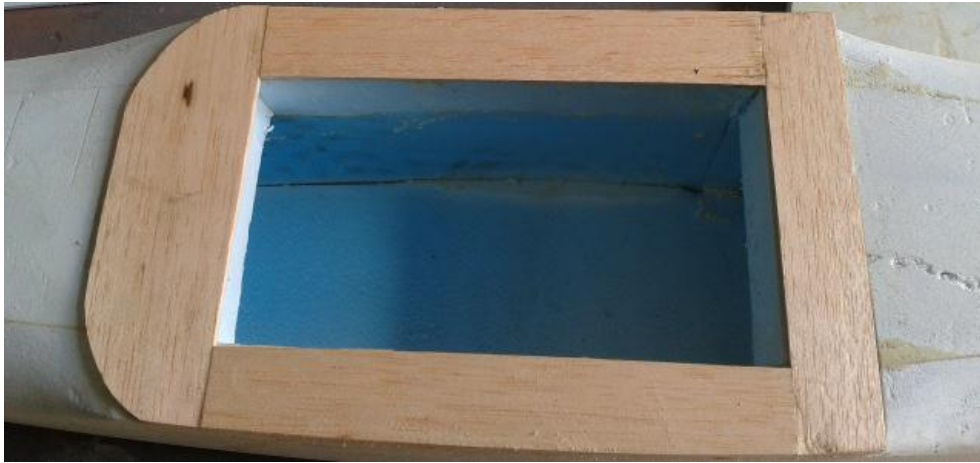


Figura 3.6: Visualização do espaço interior da fuselagem.

Na Figura 3.7 o bordo de ataque das asas tinha sido reforçado com uma tira de cartolina e já se preparava a colagem de ambos os lemes. Ao mesmo tempo esta figura dá uma ideia da dimensão da fuselagem e da envergadura de asas do aeromodelo.



Figura 3.7: Preparação da colagem dos lemes.

O próximo passo foi a colagem dos lemes de direção e profundidade utilizando a espuma de poliuretano. Como foi referido anteriormente foram feitas quase sempre duas tarefas ao mesmo tempo, e desta vez procedeu-se à construção do trem de aterragem que consiste numa vareta de aço dobrada num torno mecânico, de maneira a obter a forma desejada. Depois foi colado com a cola Araldite a uma placa de contraplacado, que mais tarde foi aparafusada noutra placa de contraplacado embutida na parte de baixo da fuselagem, restando apenas adicionar as rodas de forma a ter o aspeto da Figura 3.8.



Figura 3.8: Colagem dos lemes, e fixação do trem de aterragem.

Na Figura 3.9 podemos ver o suporte em contraplacado para fixação do motor na parte dianteira do aeromodelo colado com espuma de poliuretano e cola para madeira, a cruzeta de fixação do motor já aparafusada, e também a hélice já apertada.



Figura 3.9: Fixação do suporte do motor, do motor e da hélice.

Para finalizar o processo de construção do aeromodelo foram introduzidas ainda algumas modificações, como os suportes cilíndricos para a fixação das asas com elásticos, que atravessam a fuselagem de um lado ao outro, e os suportes para fixação dos servomotores. Depois foram adicionadas as *clevises* e as varetas de alumínio de modo a controlar os lemes de profundidade e direção, e finalmente foi aplicada uma pintura com tinta de água, de modo a que não fosse corrosiva para o material de construção e também aumentasse visibilidade do aeromodelo em voo.

Na Figura 3.10 podemos visualizar o modelo já pronto a voar, já o *spinner* incluído.





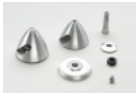




Figura 3.10: Exterior do aeromodelo completo.

### 3.3 Implementação dos componentes no aeromodelo

#### 3.3.1 Instalação dos componentes comuns

A tabela seguinte apresenta os componentes comuns necessários para os vários modos de controlo remoto do AMT:

Descrição	Imagem	Quantidade
Bateria Li-po Turnigy 5000 mah		1
Variador brushless 50 Amperes		1
Motor brushless GT2826/04-752 W		1
Hélice 11x5.5E		1
Spinner Alu/40mm-5mm		1
Servomotor Savox		2
Extensão JR/Gr 60 cm		2

Alguns dos componentes comuns e também alguns dos componentes dos subcapítulos que se seguem, precisam de suportes de fixação no interior do AMT, então foram construídos os seguintes suportes:



Figura 3.11: Suporte para a bateria *Li-Po* e para a bateria de 4.8 V que alimenta o HUB USB.



Figura 3.12: Suporte para o recetor rádio, controlador de servos, circuito do *multiplexer*, e HUB USB.



Figura 3.13: Instalação dos suportes no interior do AMT.



Figura 3.14: Suporte para fixação do SBC Snowball.

### 3.3.2 Instalação dos componentes para o modo de controlo clássico

Para o modo de controlo clássico utilizam-se os componentes comuns referidos no ponto anterior. Aqui o foco vai ser na forma com que os componentes eletrónicos se interligam. A bateria Li-Po (*lithium-ion polymer battery*) de 11.7 volt alimenta o variador, também conhecido como ESC (*electronic speed controller*) que tem como funções controlar a rotação do motor e alimentar o recetor rádio de seis canais da Figura 3.15 através de um BEC (*battery elimination circuit*) que alimenta o recetor rádio a 4.8 volt. O próximo passo é ligar os servomotores e o ESC nos respetivos canais, de modo a converter os sinais PWM do recetor em movimento angular nos casos dos servomotores e em rotação do motor principal através do ESC.



Figura 3.15: Recetor rádio de 6 canais.

A fixação do recetor rádio no suporte e as ligações dos servomotores e do ESC estão demonstradas na Figura 3.16.

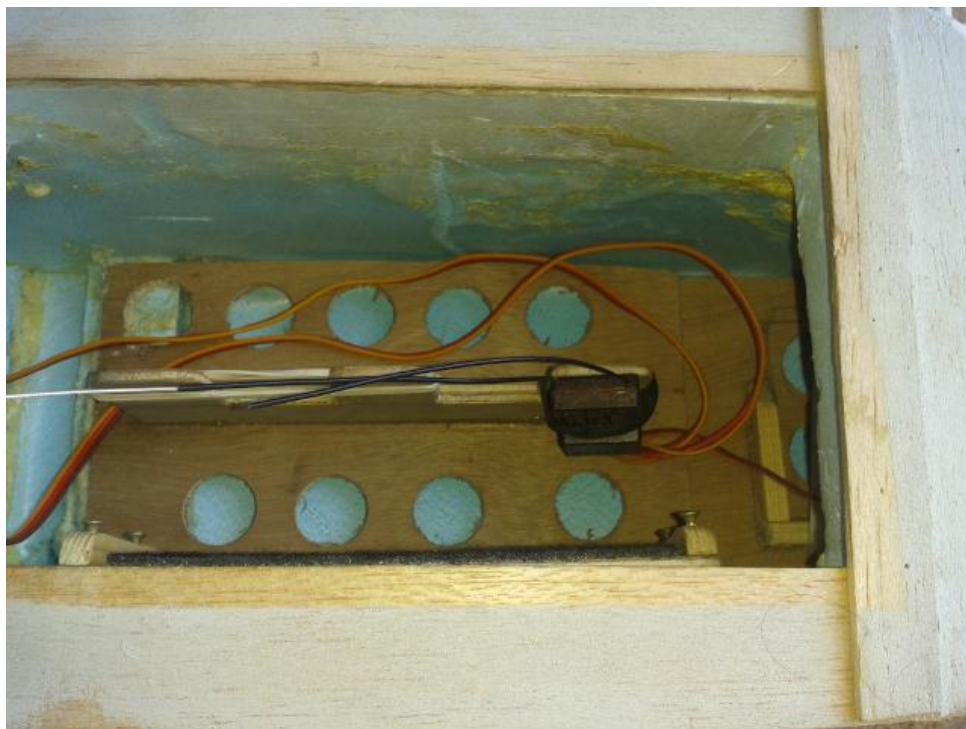


Figura 3.16: Fixação do recetor rádio já com os servomotores e o ESC ligados.

### 3.3.3 Instalação dos componentes para o modo de controlo por WLAN e WWAN

A tabela seguinte apresenta os componentes necessários para o modo de controlo remoto por WLAN e WWAN:

Tabela 3.2: Componentes para o controlo remoto por WLAN e WWAN

Descrição	Imagem	Quantidade
SKY-S9500-ULP-C12 (Snowball)		1
Bateria Li-Ion 3.7V 2600mAH		1
Pololu Micro Maestro 6- Channel USB Servo Controller		1
HUB de 7 portas alimentado		1
Bateria 4.8 V 2500mAH para o HUB USB		1
Adaptador OTG Mini USB A- USB A-Socket		1
Multiplexer Quad 2:1		1
Condensador cerâmico de 1nF		3
Logitech quickcam Pro		1
Modem 3G MF636		2
Cola Epoxy		1

Neste tipo de controlo remoto foi implementado o sistema de segurança que faz a troca com o com controlo clássico em caso de o sinal *WLAN* ou *WWAN* estar fraco, ou em caso de troca manual. Para isso foi implementado o circuito da Figura 3.17, que é controlado de acordo com a Tabela 3.3 através de um sinal de controlo proveniente de um dos canais do microcontrolador de servos, que ou está a cinco volt, ou a zero volt.

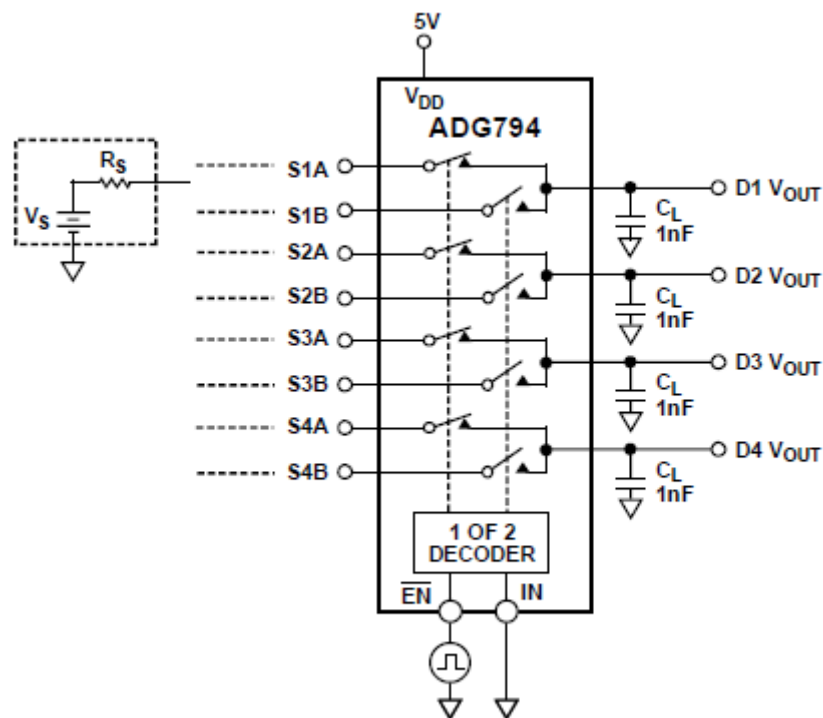


Figura 3.17: Circuito de multiplexagem utilizado (13).

Tabela 3.3: Tabela de verdade do *multiplexer* (13).

$\overline{\text{EN}}$	IN	D1	D2	D3	D4	Function
1	X	High-Z	High-Z	High-Z	High-Z	Disable
0	0	S1A	S2A	S3A	S4A	IN = 0
0	1	S1B	S2B	S3B	S4B	IN = 1

Na implementação prática do circuito de multiplexagem só foram precisos três canais: para o leme de profundidade, para o leme de direção, e para o acelerador. O circuito ficou com o aspeto da Figura 3.18, já depois de ter sido aplicada cola epoxy para uma melhor fixação dos fios soldados nas pistas de cobre.

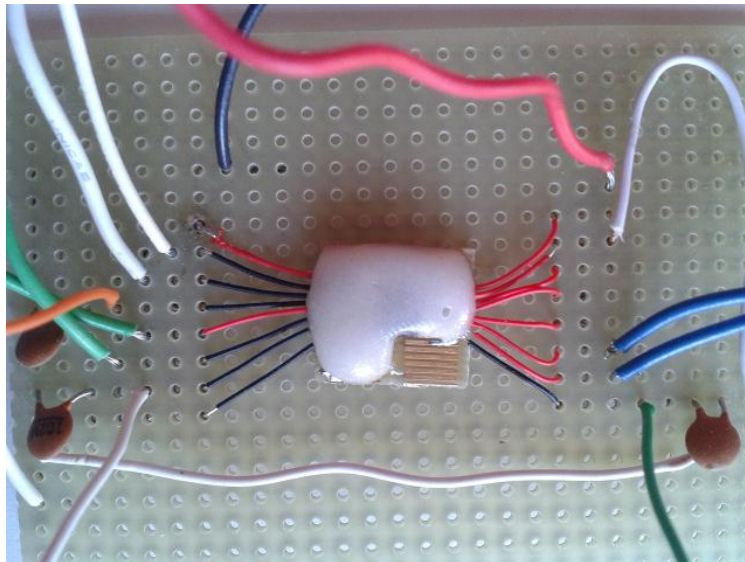


Figura 3.18: Implementação do circuito de multiplexagem.

O sistema está agora completo, pelo que é possível fazer o esquema de ligações da Figura 3.19, em que a bateria Li-Po de 11.7 volt alimenta o ESC que controla o motor, e contém um BEC que alimenta o recetor rádio, o microcontrolador de servos (Pololu), e o *multiplexer* a 4.8 volt.

Os sinais PWM SxA são provenientes do recetor rádio e os sinais PWM SxB são provenientes do microcontrolador Pololu. O último canal do Pololu (em baixo) está configurado como *output* e serve de *flag* de controlo para a entrada IN do *multiplexer*, se estiver a zero volt os sinais nas saídas D1, D2, D3, vêm do recetor rádio, se estiver a cinco volt, o sinal do Pololu será encaminhado para as mesmas saídas.

Todos os periféricos estão ligados por cabos USB (com os devidos adaptadores) ao HUB USB que tem a sua própria bateria de 4.8 volt. O SBC Snowball recebe os dados de controlo do cliente e interage com todos os periféricos, garantindo o funcionamento do sistema.

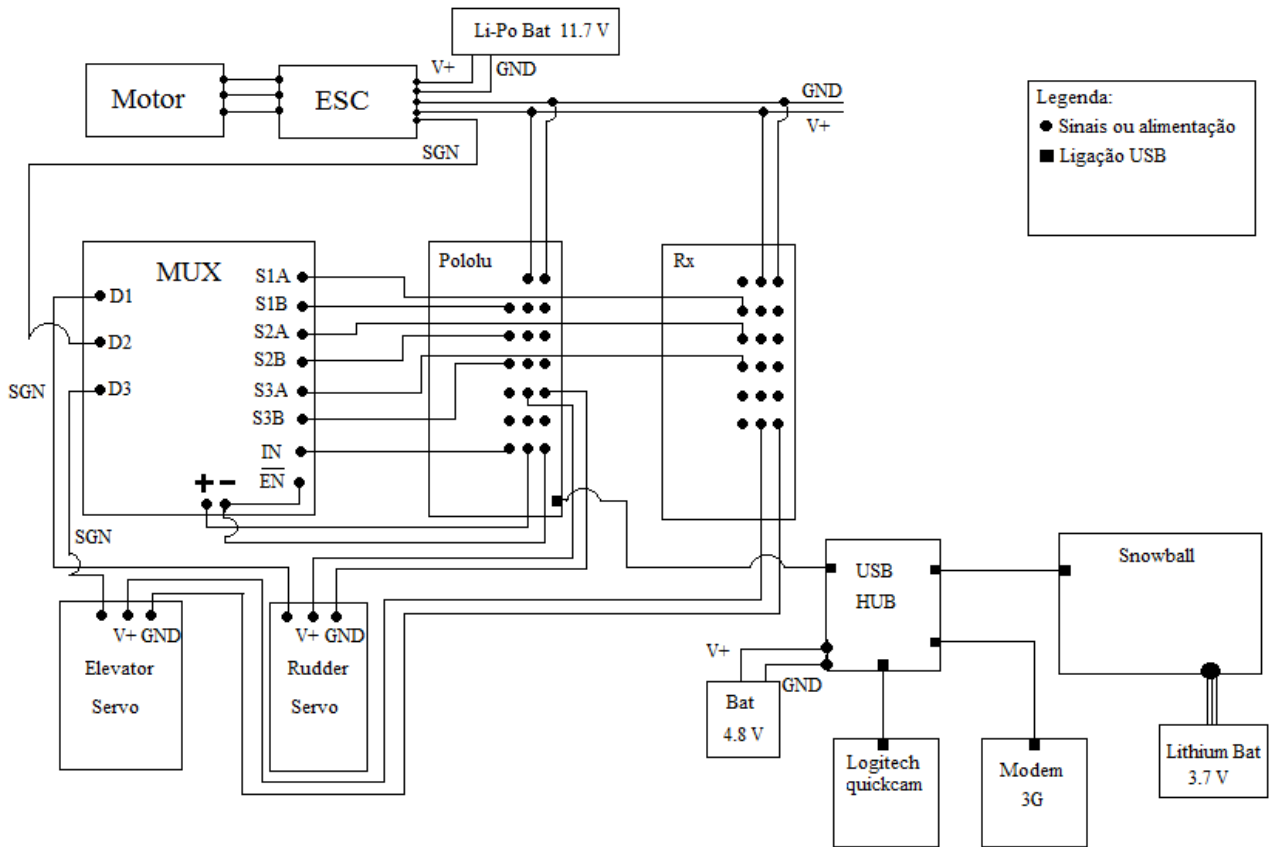


Figura 3.19: Sistema de ligações completo.

## 4. Análise do sinal de *pulse width modulation*

### 4.1 Utilizando o recetor rádio

No caso do sinal recebido no recetor rádio podemos verificar que para todos os canais utilizados a largura do pulso varia entre 1 ms e 2 ms, enquanto a amplitude mantém-se constante nos 3.6 volt. Já o período de repetição dos pulsos é ligeiramente superior a 20 ms.



Figura 4.1: Pulso mínimo (esquerda), Pulso máximo (direita) (Escala: X- 1 ms, Y- 2 volt).

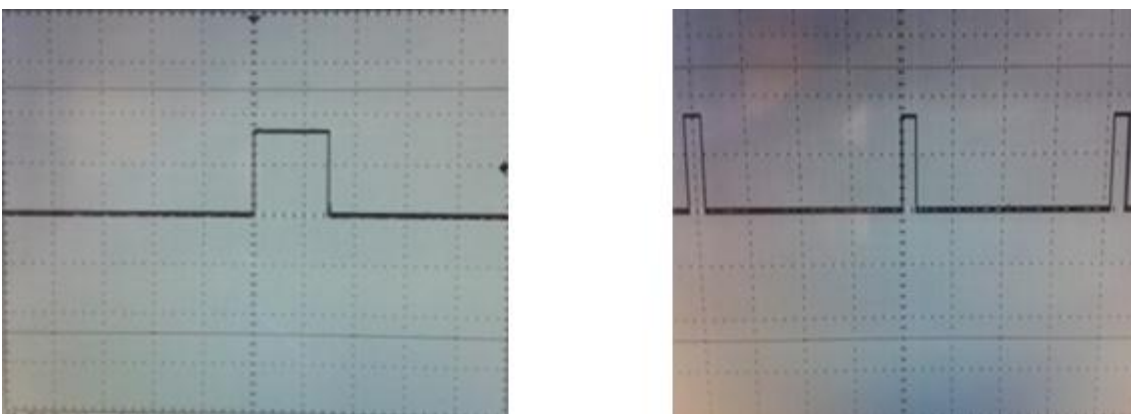


Figura 4.2: Pulso médio (esquerda) (Escala: X- 1 ms, Y- 2 volt), Período de repetição de pulsos (direita) (Escala: X- 5 ms, Y- 2 volt).

## 4.2 Utilizando o microcontrolador de servos Pololu

No caso do sinal gerado pelo microcontrolador de servos, podemos verificar que para todos os canais utilizados a largura do pulso varia entre 1 ms e 2 ms, enquanto a amplitude se mantém constante nos 4.5 volt. Apesar da alimentação nominal dos servomotores ser de 5 volt, a variação de amplitude verificada nos pulsos do recetor rádio e do microcontrolador Pololu não tem influência no movimento dos servomotores, pois a tolerância destes é bastante grande. Já o período de repetição dos pulsos é 20 ms.

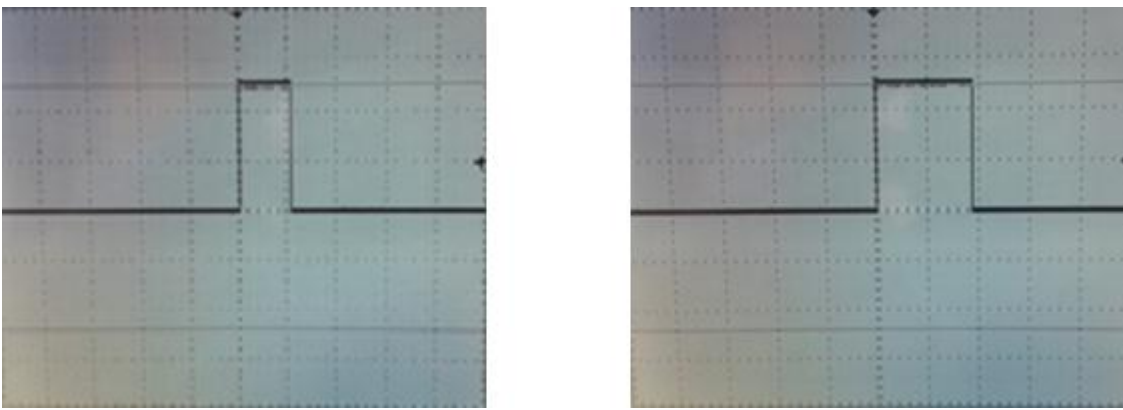


Figura 4.3: Pulso mínimo (esquerda), Pulso máximo (direita) (Escala: X- 1 ms, Y- 2 volt).

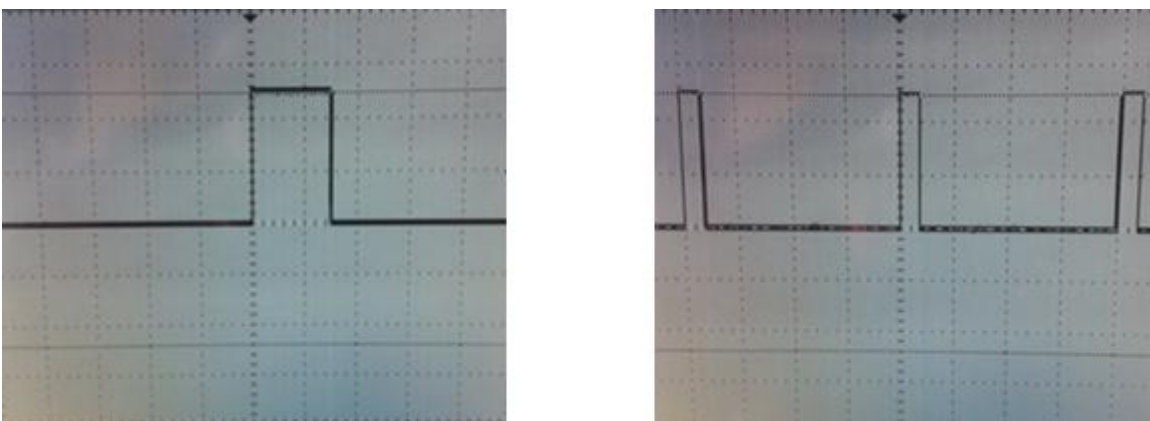


Figura 4.4: Pulso médio (esquerda) (Escala: X- 1 ms, Y- 2 volt), Período de repetição de pulsos (direita) (Escala: X- 5 ms, Y- 2 volt).

### 4.3 Utilizando multiplexagem dos dois modos referidos anteriormente

No caso em que a *flag* de controlo está no modo *high* (cinco volt) o *multiplexer* seleciona o sinal proveniente do microcontrolador de servos e desta forma do computador de bordo, o *snowball*. Como se pode observar pelas figuras abaixo as características de largura do sinal mantêm-se inalteradas, notando-se uma diminuição na amplitude para quatro volt e apenas um ligeiro aumento do ruído na linha dos zero volt, que não afeta a performance do sistema.

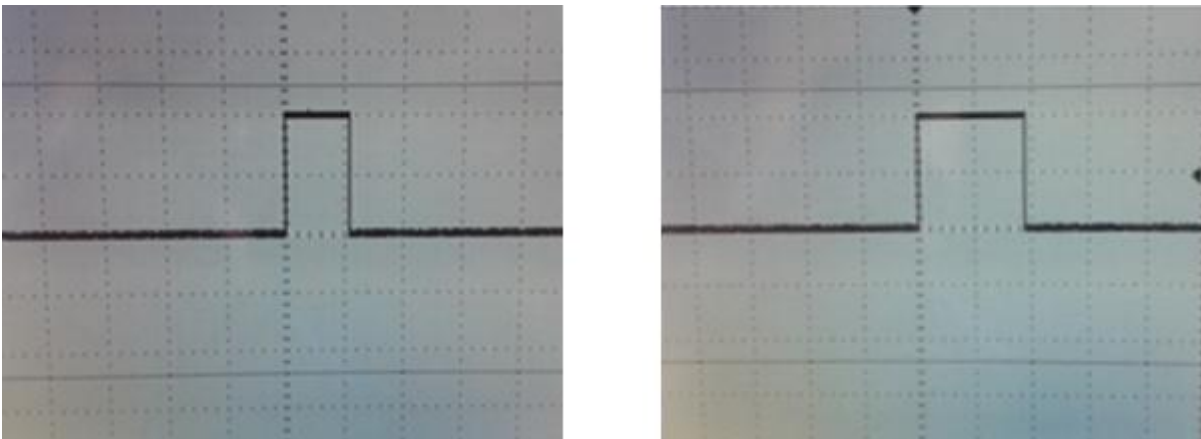


Figura 4.5: Pulso mínimo (esquerda), Pulso máximo (direita) (Escala: X- 1 ms, Y- 2 volt).

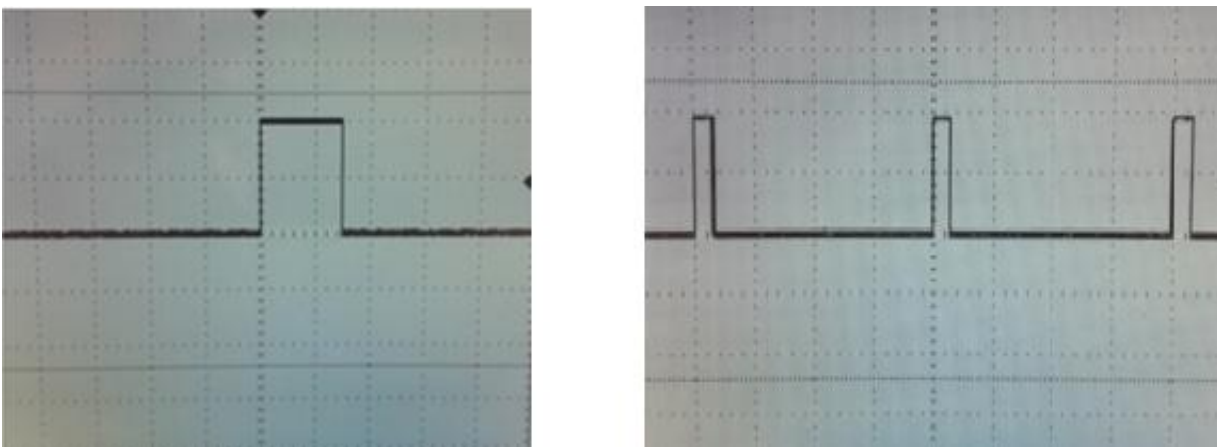


Figura 4.6: Pulso médio (esquerda) (Escala: X- 1 ms, Y- 2 volt), Período de repetição de pulsos (direita) (Escala: X- 5 ms, Y- 2 volt).

No caso em que a *flag* de controlo está no modo *low* (zero volt) o *multiplexer* seleciona o sinal proveniente do recetor rádio. Como se pode observar pelas figuras abaixo as características do sinal mantêm-se inalteradas, notando-se apenas um ligeiro aumento do ruído na linha dos zero volt, que não afeta a performance do sistema.

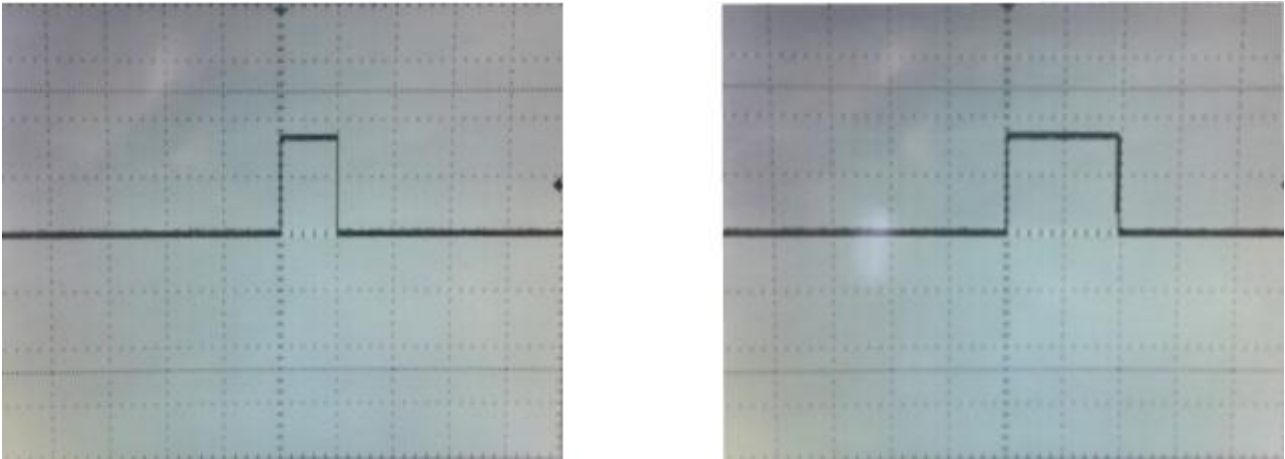


Figura 4.7: Pulso mínimo (esquerda), Pulso máximo (direita) (Escala: X- 1 ms, Y- 2 volt).

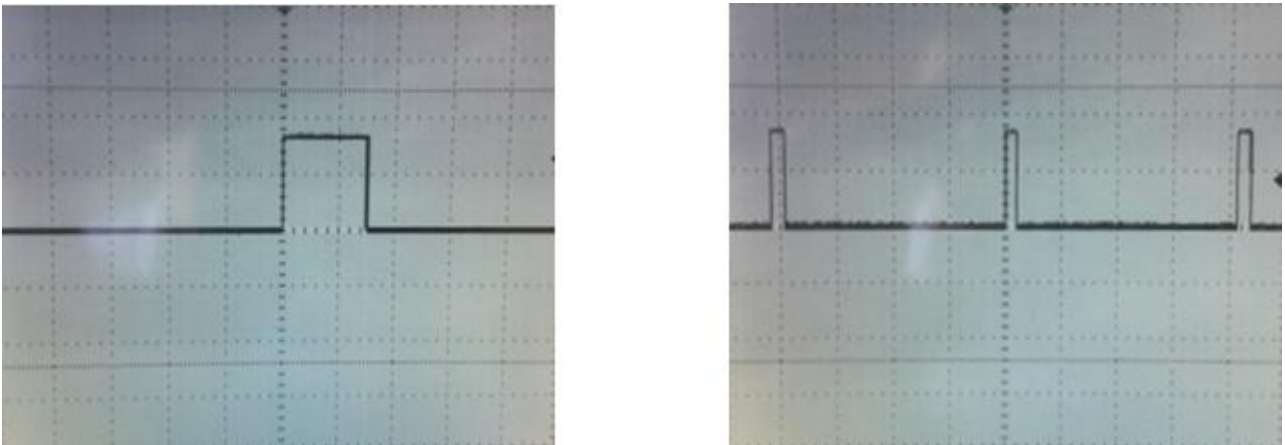


Figura 4.8: Pulso médio (esquerda) (Escala: X- 1 ms, Y- 2 volt), Período de repetição de pulsos (direita) (Escala: X- 5 ms, Y- 2 volt).

# 5. Tecnologias de *software*

## 5.1 Modelo Cliente Servidor

O modelo cliente servidor consiste na comunicação entre programas clientes e programas servidores. Um programa cliente corre num terminal (máquina que se encontra na periferia da rede) que pede e recebe um serviço de outro programa servidor (que normalmente corre em *background*) que se encontra noutro terminal (14). Tipicamente o programa cliente e o programa servidor não se encontram no mesmo terminal, isto é, usam uma rede de computadores que pode aceder à internet para comunicar entre si, mas também podem residir no mesmo sistema. Para efetuar a comunicação entre clientes e servidores é utilizado pelo menos um dos protocolos da camada de transporte: o UDP (*user datagram protocol*) que não é orientado à conexão nem garante entrega fiável de dados, ou o TCP (*transmission control protocol*) que é orientado à conexão e garante entrega fiável de pacotes de dados, entre outras características. Perante as características dos protocolos foi decidido optar pela utilização do protocolo TCP nesta dissertação, devido à sua maior fiabilidade, encontrando maneiras de atenuar a eventual perda de dados através da implementação de um *timeout* num processo do servidor descrito num capítulo posterior.

## 5.2 *Live streaming media*

O termo *media* refere-se a multimédia, isto é, a vários tipos de conteúdos (vídeo, áudio, texto) que podem ser tratados como um só.

*Streaming* é o método como esta informação multimédia é distribuída de um servidor para um ou mais clientes através do meio de transmissão, que permite a receção e visualização num *mediaplayer* por exemplo, de média que está continuamente a chegar, sem que seja necessário ter um ficheiro completo no cliente para visualizá-lo, aliás normalmente o leitor não grava a informação recebida, esta costuma ser descartada.

A internet hoje em dia é dos meios mais utilizados para transmitir *live streaming media*, que consiste em pacotes de dados que são encaminhados por protocolos de transporte como o UDP, o RTSP (*Real-time Streaming Protocol*), o *HTTP adaptive bitrate streaming*, ou até mesmo o TCP que requer algum trabalho de *software* para que não existam atrasos, dado que o protocolo não descarta pacotes e reenvia pacotes perdidos.

No âmbito desta dissertação apenas o *stream* de vídeo foi implementado, utilizando a aplicação Motion para difusão, cujo funcionamento será descrito num capítulo posterior, e o leitor de vídeo que foi implementado no programa cliente para reproduzir este *stream*.

### **5.3 *Dynamic Domain Name Server***

Os ISPs (*Internet Service Providers*) normalmente fornecem a uma máquina um endereço ip real (ip visível por qualquer computador ligado à internet) que não é fixo, isto é, muda num determinado período de tempo pré-definido. O DDNS (*Dynamic Domain Name Server*) foi criado para resolver este problema, associando um nome de um subdomínio ao endereço ip da conexão do utilizador. Assim a partir do momento que este nome é registado e o programa de DDNS é instalado e executado na máquina que queremos saber o endereço ip, qualquer mudança no endereço ip feita pelo ISP é atualizada para um servidor de nomes que nos registámos anteriormente (15).

# 6. Implementação do controlo do aeromodelo

## 6.1 *Graphical User Interface* do cliente

O controlo do AMT é efetuado no cliente através de um GUI (graphical user interface) implementado em Java, que consiste num leitor de vídeo incorporado na janela principal (no centro) que lê o *stream* de vídeo proveniente do serviço Motion, um programa de captura e distribuição de vídeo (será descrito mais à frente neste capítulo) que corre no servidor. O GUI dispõe de *scrollers* (em baixo) que indicam as posições de movimento angular dos lemes no caso do *Rudder*, *Elevator* e *Aileron*, e a percentagem de aceleração do motor principal no caso do *Throttle*. O GUI consiste ainda em três barras (em cima à direita) que nos dão a informação sobre a força do sinal no caso de o AMT estar ligado a uma rede local sem fios, a uma rede de telecomunicação celular, ou às duas em simultâneo, tendo o utilizador que escolher a qual das redes se vai ligar previamente. A terceira barra refere-se à carga da bateria *Li-ion* que alimenta o SBC a bordo do AMT. O GUI consiste ainda em três botões (ao centro à direita) que indicam o tipo de comunicação a ser utilizado, se o cliente está ligado ao servidor através de uma rede local sem fios, de uma rede de telecomunicação celular, ou se está ligado via rádio frequência. Estes botões são controlados pela *flag* de controlo que permite passar do modo WLAN e WWAN para o modo clássico, quer seja em caso de troca manual (através de uma combinação de teclas, se o controlo for efetuado pelas teclas, ou por um botão do *joystick* caso o controlo seja efetuado pelo *joystick*). O modo de controlo encontra-se representado por dois botões (em baixo à direita) que indicam se o AMT está a ser controlado pelo teclado, ou pelo *joystick*.

Abaixo são apresentadas a Figura 6.1 que nos mostra o GUI quando o AMT é controlado através de uma WLAN ou WWAN, e a Figura 6.2 que nos mostra o GUI quando se passa o controlo para o modo clássico. Neste caso o vídeo continua a ser transmitido, pois o serviço de vídeo é separado do programa servidor.

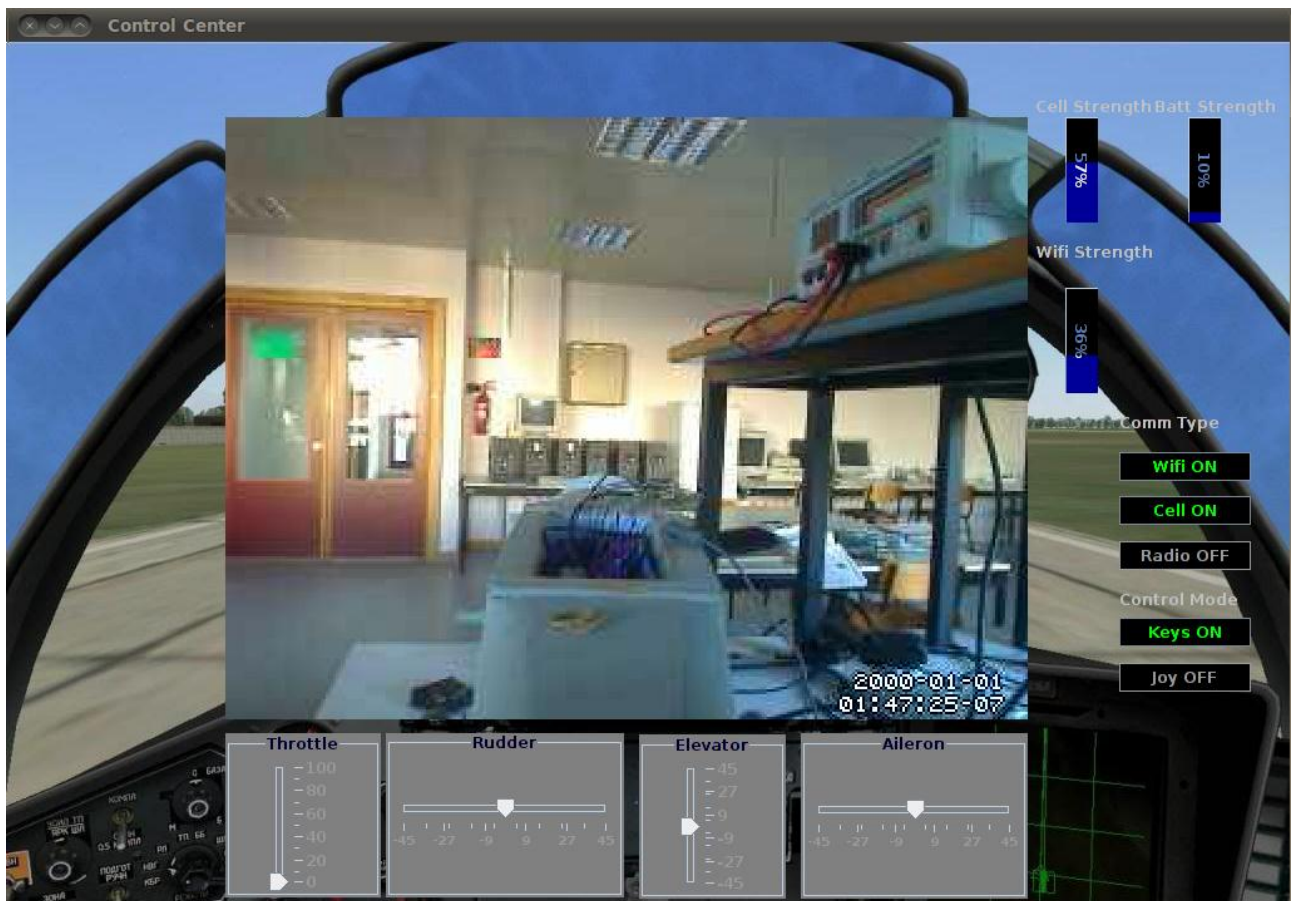


Figura 6.1: GUI com comunicação por WLAN ou WWAN

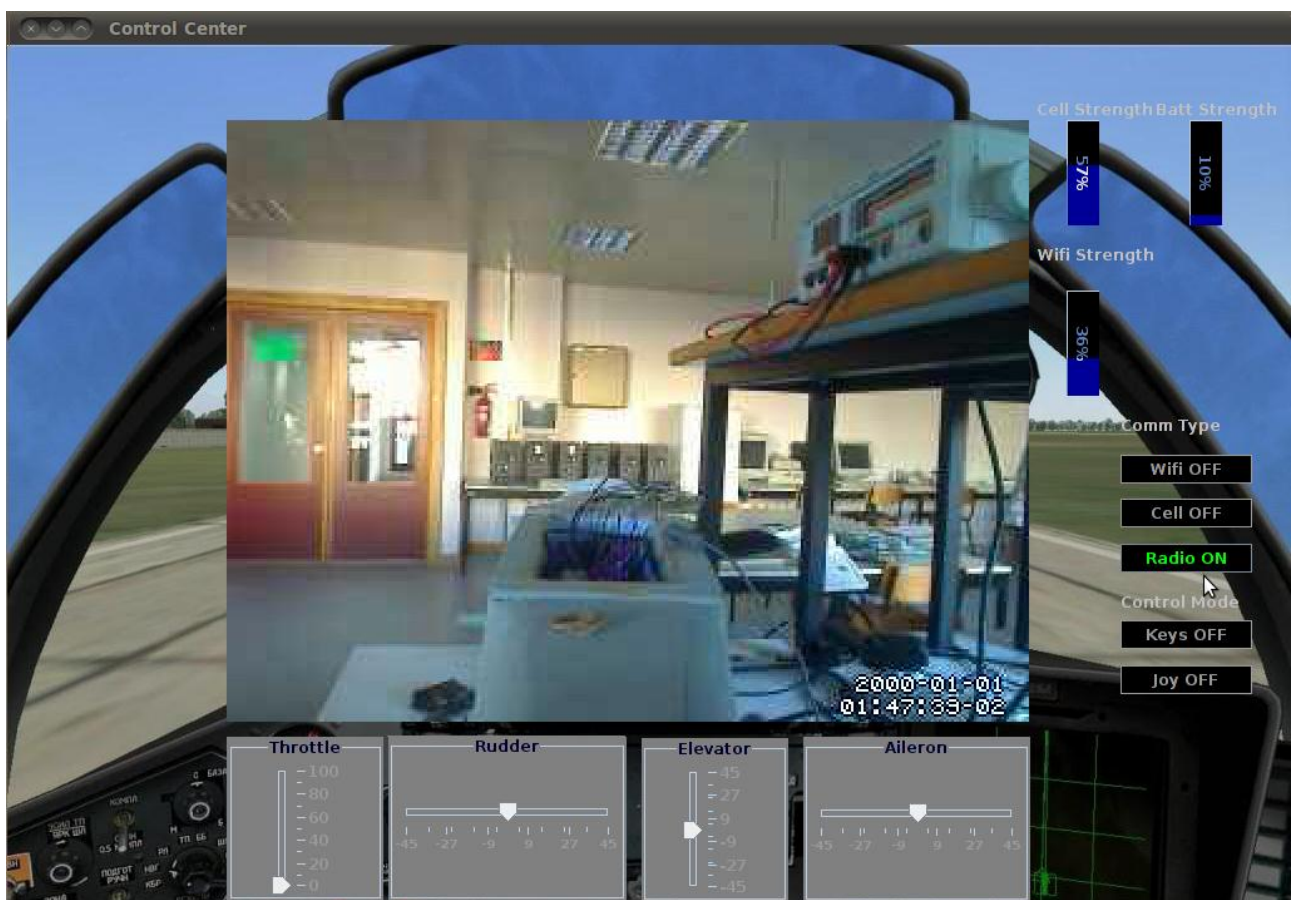


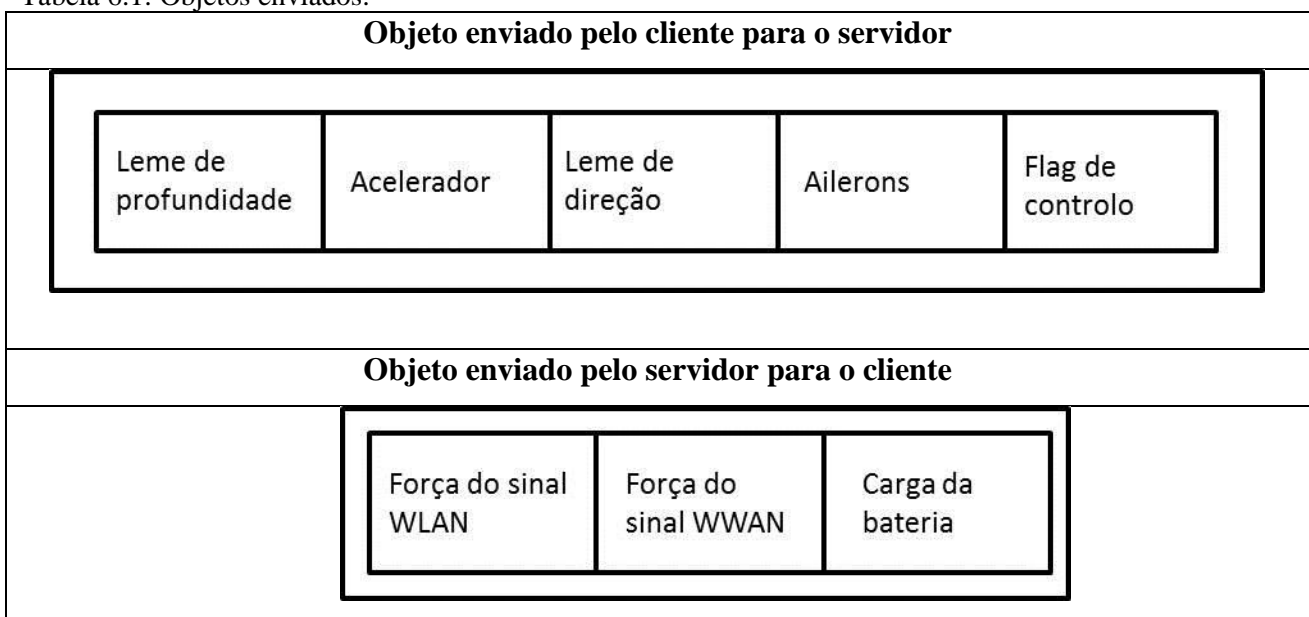
Figura 6.2: GUI com comunicação através do modo clássico

## 6.2 Comunicação entre cliente e servidor

A comunicação entre o cliente e o servidor é estabelecida através de BSD *sockets*, usando o protocolo TCP/IP, seguindo o modelo do ponto 5.1, estando o servidor instalado no SBC Snowball e o cliente num computador remoto. O servidor inicia a comunicação ficando à espera que um cliente se ligue num endereço e porto pré-definidos, e só aceita um cliente a cada ligação estabelecida. Depois de a ligação ser estabelecida, a comunicação é feita através de um protocolo de comunicação bidirecional que consiste em dados encapsulados em objetos.

Abaixo encontra-se a Tabela 6.1 que contém a representação dos objetos.

Tabela 6.1: Objetos enviados.



Existem dois tipos de objetos, o enviado pelo cliente para o servidor que consiste em cinco números inteiros de 64 *bits* referentes ao acelerador, ao leme de direção, ao leme de profundidade, aos ailerons (apesar de o AMT não ter ailerons foi implementada esta funcionalidade para se poder utilizar noutros aeromodelos), e à *flag* que comanda o modo de comunicação utilizado para controlar o AMT, tendo esta apenas dois valores 100 ou 200, escolhidos de acordo com a API do microcontrolador de servomotores para gerar zero ou cinco volt na saída respetiva, permitindo assim efetuar a troca do modo WLAN e WWAN para o modo clássico.

O tipo de objeto enviado do servidor para o cliente consiste em três inteiros que representam as forças dos sinais WLAN, WWAN e da bateria *Li-ion* do AMT. Através destes dados é possível fazer a deteção do tipo de comunicação utilizado, que vimos no GUI do cliente.

### 6.3 Descrição do funcionamento do cliente.

O cliente é responsável por disponibilizar serviços de escrita de dados de controlo para o servidor, bem como de leitura de dados do servidor com informação para o GUI. O funcionamento geral é dado pelo diagrama da Figura 6.3, que mostra como as tarefas interagem entre si. Seguidamente apresenta-se a descrição das tarefas e quais as suas funções.

Após a conexão ser estabelecida entre o cliente e o servidor, a tarefa principal faz a seleção do modo de controlo a ser utilizado, joystick caso esteja um ligado a uma porta USB, ou o teclado se não estiver nada conectado, e de seguida inicializa as tarefas periódicas de escrita de dados de controlo para o servidor, de leitura de dados de informação do servidor, e de atualização do GUI. A tarefa periódica de leitura de teclas implementa a classe `KeyListener`, esta contém os métodos `keyPressed` e `keyReleased` que detetam eventos quando as teclas são pressionadas e libertadas respetivamente.

Abaixo está a definição dos métodos:

```
public synchronized void keyPressed(KeyEvent e){  
/*teclas pressionadas*/}
```

```
public synchronized void keyReleased(KeyEvent e){  
/*teclas que foram libertadas*/}
```

Com recurso a estes métodos foi implementado um algoritmo que deteta o tempo que as teclas são pressionadas e faz a atualização das variáveis de controlo do AMT com um período de 60 (ms).

No caso da tarefa de leitura do *joystick* não é necessário fazer uma monitorização periódica dos eixos e dos botões, pois esta já é feita pela API correspondente. Basta apenas guardar a informação em variáveis que são acedidas pelas tarefas periódicas de atualização do GUI e de escrita de dados de controlo. A API em Java utilizada para o acesso e monitorização do *joystick* é a `JInput` (16).

As tarefas periódicas de atualização do GUI e de escrita de dados de controlo acedem a estas variáveis com os períodos de 60 (ms) e 80 (ms) respetivamente. A tarefa periódica de leitura de dados com informação do estado das redes e bateria do Snowball corre com o período de 3 (s) e guarda a informação noutras variáveis para depois serem acedidas pela tarefa de atualização do GUI. Abaixo encontra-se a esquematização no diagrama da Figura 6.3.

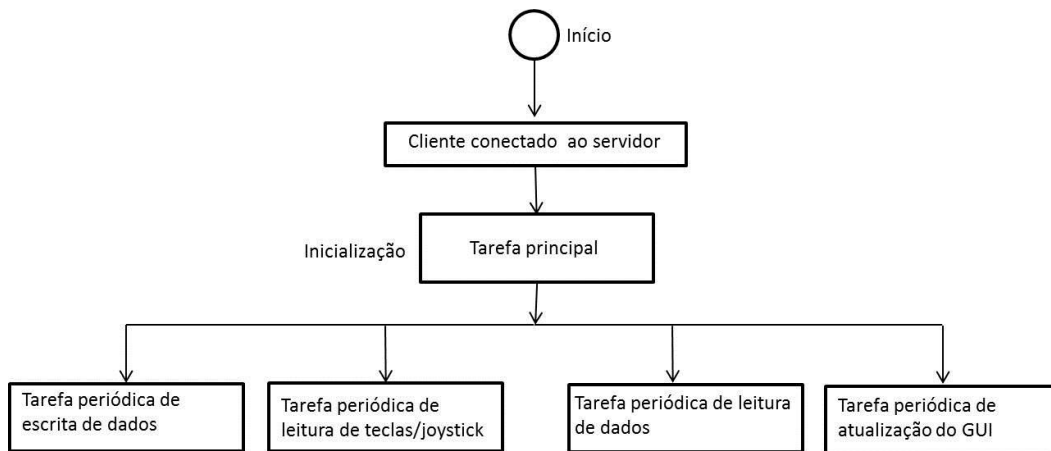


Figura 6.3: Diagrama das tarefas executadas no cliente.

## 6.4 Descrição do funcionamento do servidor

No arranque do servidor são inicializados todos os serviços necessário para a comunicação com o cliente, nomeadamente o serviço de vídeo Motion e o serviço de controlo e monitorização do AMT, que é inicializado pela tarefa principal da aplicação em Java desenvolvida, como se pode observar na Figura 6.4.

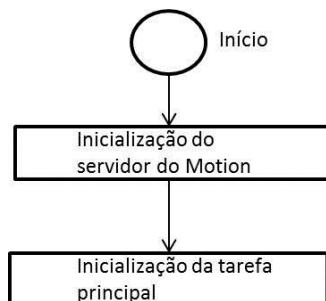


Figura 6.4: Diagrama do servidor.

A tarefa principal inicia o socket que espera até que um cliente se ligue no endereço ip e porto correto, se o *socket* não receber dados durante três segundos gera uma exceção de *timeout*, reinicia a

ligação e passa para o modo de controlo clássico. Caso o *socket* receba dados sem interrupção, de seguida inicializa o serviço de controlo e monitorização, que consiste em duas tarefas periódicas, uma de leitura e uma de escrita, de forma a se poder ter um período curto para os dados de controlo recebidos e um período mais longo de tempo para a monitorização do AMT. Estes períodos são de 80 ms para a tarefa de leitura de dados de controlo e de 3 s para a tarefa de escrita com informação das redes e da bateria do Snowball. O funcionamento da tarefa principal encontra-se representado abaixo na Figura 6.5.

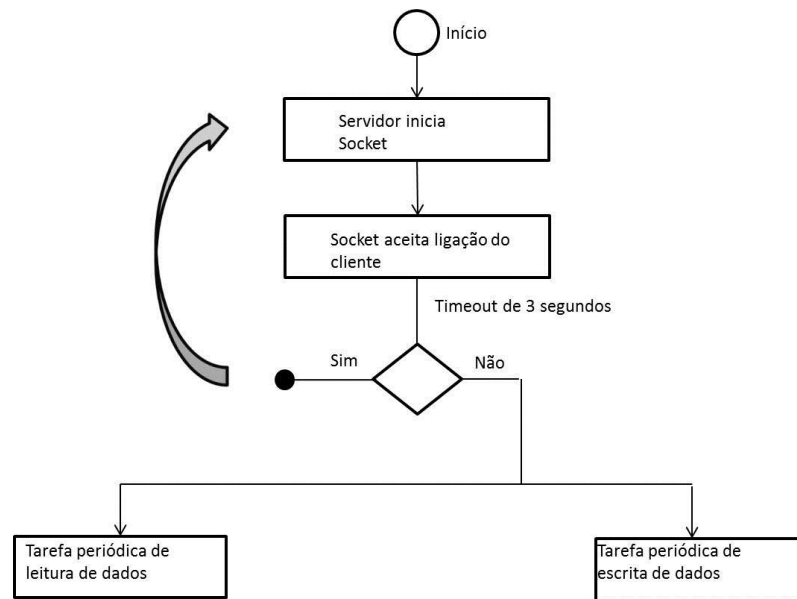


Figura 6.5: Diagrama da tarefa principal.

## **6.5 *Video streaming***

### **6.5.1 Motion**

O Motion é uma aplicação que permite a visualização de vídeo em tempo real captado por várias câmaras, através do seu servidor web que comunica no porto 80, bem como o armazenamento de vídeo e imagens em vários formatos. Tem ainda a capacidade de detetar movimento através dos pixéis que mudam de uma imagem para outra. Para além das funcionalidades já referidas falta referir um aspeto de grande importância para esta dissertação, que é a baixa latência do stream de vídeo, tornando a visualização do vídeo quase de tempo real. Todas estas ações são executadas utilizando o driver video4linux2 que permite a interação com câmaras compatíveis, em que o dispositivo é detetado em `/dev/videox`.

O Motion apesar de não ter um ambiente gráfico, é extremamente configurável através do seu ficheiro de configuração que permite alterar entre outras coisas, a resolução da câmara, a qualidade das imagens, o período com que se guardam imagens, e o número de câmaras. Infelizmente na versão atual do sistema operativo do Snowball, a resolução do Motion está limitada a 320x240, mas mesmo assim é suficiente para a visualização do ambiente exterior ao AMT, permitindo o seu controlo.

A escolha do Motion como aplicação de captura de vídeo teve ainda base na dissertação de Aldric Trindade Négrier (17), que já tinha testado outras alternativas e verificado que esta é uma aplicação com muitas vantagens em aplicações deste tipo. Todos estes atributos fazem do Motion uma aplicação ideal para vídeo vigilância ou monitorização de espaços exteriores, como é o caso.

# 7. Ensaio do aeromodelo

## 7.1 Condução remota do aeromodelo

Como todo o aeromodelo foi construído de raiz, foram feitos vários ensaios para aperfeiçoar aspectos de aerodinâmica como o posicionamento do motor e das asas no ângulo certo, e o nivelamento dos lemes. A perspectiva escolhida para o AMT foi a de terceira pessoa pois esta permite ter uma melhor noção do espaço em redor e da direção na qual o aeromodelo se movimenta.

A autonomia do AMT é de cerca de 30-35 minutos superando assim a autonomia de aviões RTF (*ready to fly*), ou ARF (*almost ready to fly*) disponíveis no mercado, que atingiam no máximo cerca de 25 minutos e não garantiam o espaço de carga necessário.

## 7.2 Condução remota utilizando WLAN

A condução remota utilizando uma WLAN só foi utilizada em laboratório, mas esta serviu para testar a comunicação entre cliente e servidor e garantir que a latência da rede não era muito elevada, cerca de 20 ms em média. A média de tempo foi calculada com base em tempos medidos com o comando ‘ping ip do cliente’. Fica assim provado que com uma infraestrutura WLAN mais alargada, por exemplo com a instalação de antenas omnidirecionais, a comunicação seria até de menor latência do que numa WWAN, oferecendo assim uma melhor experiência de voo sem custos para o utilizador.

## 7.3 Condução remota utilizando uma WWAN

A condução remota utilizando uma WWAN foi efetuada em vários ensaios, em dois terrenos diferentes, numa tentativa de encontrar uma área com cobertura HSPA, e assim permitir o controlo do AMT numa zona alargada e fora de linha de vista de acordo com os objetivos propostos para esta dissertação.

No terreno o computador cliente estava conectado através de outro modem 3G, o que fez com que a latência subisse para uma média de 160 ms, mas com picos de latência que por vezes ultrapassavam 1 s, tornando a experiência de condução por vezes um pouco difícil. A média de tempo foi calculada com base em tempos medidos com o comando ‘ping ip do cliente’.

Nos ensaios o controlo do AMT foi efetuado no chão com sucesso, mas derivado das oscilações que a potência do sinal da rede de telecomunicação celular sofreu, a comunicação foi muitas vezes perdida, o que levou a que a condução com o AMT em voo fosse efetuada somente através do modo clássico.

## 7.4 Comparação entre os três modos de controlo

A experiência de condução do AMT varia em cada um dos modos de controlo. Cada um dos modos de controlo apresenta vantagens e desvantagens na sua utilização. A Tabela 7.1 ilustra esta situação.

Tabela 7.1: Vantagens e desvantagens dos modos de controlo.

<b>Modo de controlo</b>	<b>Vantagens</b>	<b>Desvantagens</b>
<b>Clássico</b>	-Praticamente sem latência no controlo.	-O controlo tem de ser feito em linha de vista. Pequena cobertura.
<b>WLAN</b>	-Baixa latência no controlo. -Sem custos para o utilizador. -O controlo pode não ser feito em linha de vista. -Qualquer PC com o programa cliente pode aceder ao SBC.	-Pequena cobertura. -Precisa de um PC para efetuar o controlo.
<b>WWAN</b>	-O controlo pode não ser feito em linha de vista. -Vasta área de cobertura. -Qualquer PC com o programa cliente pode aceder ao SBC.	- Precisa de um PC para efetuar o controlo. -Nota-se alguma latência no controlo. -Tem custos para o utilizador.

## 8. Conclusões e trabalho futuro

Toda a estrutura do AMT revelou-se bastante apropriada para a sua aplicação, tendo efetuado vários ensaios em modo clássico para aperfeiçoar a aerodinâmica, e mais tarde o ensaio no modo WWAN.

A implementação de código desta dissertação foi feita numa linguagem orientada a objetos que além de estar no âmbito do curso, tem o intuito de tornar a aplicação o mais compatível possível com o maior número de sistemas operativos, estou-me a referir à linguagem de programação Java. O GUI desenvolvido em Java, permitiu uma monitorização do AMT que tornou possível o controlo em modo WLAN e WWAN, cumprindo os objetivos propostos para esta dissertação, apesar de a condução em modo WWAN não ter sido completamente bem-sucedida.

O SBC utilizado apresenta algumas falhas a nível da arquitetura da comunicação USB, pois a alimentação do HUB USB causa perda de dados, e o suporte a nível de software também é bastante limitado, dado nenhum dos sensores presentes se encontrava devidamente suportado.

A cobertura de redes de telecomunicação celular no Algarve ainda fica aquém das expectativas para projetos desta natureza.

O trabalho futuro passará por utilizar outro SBC que tenha especificações semelhantes ou superiores ao Snowball, mas que seja melhor suportado a nível de software. Existindo suporte para todos os sensores suportados no SBC: giroscópio, acelerómetro, GPS, barómetro, já será possível uma abordagem totalmente diferente no controlo do AMT, será possível torná-lo completamente autónomo e estabelecer rotas que sejam cumpridas com todo o rigor necessário, pois existe toda a informação necessária, a posição, direção, velocidade e altitude do AMT. Para efetuar este controlo serão avaliados diferentes algoritmos que permitirão fazer a estabilização do sistema de controlo do AMT dentro de um limite de tempo aceitável.

Outras melhorias que poderão ser implementadas são a inclusão da plataforma Java de tempo real para permitir maior precisão no controlo dos servos pelo SBC e a utilização de redes de telecomunicação celular de nova geração como o 4G, que têm uma cobertura em evolução contínua, permitindo um aumento de desempenho significativo em tempo de latência da rede, esperando-se em boas condições de cobertura ter um controlo praticamente instantâneo do AMT.

# Apêndice A: Configuração do *kernel* Linux

## Recompilação do *kernel*

Para recompilar o *kernel*, executar os seguintes comandos:

#Correr na diretoria fonte:

```
(1) make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- menuconfig
```

#Selecionar no menu:

```
Device Drivers --> Power supply class support -->
```

```
Enable support of the 9100 Li-ion battery charging
```

```
(2) make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- uImage
```

```
(3) make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- modules
```

#Substituir o *kernel* e os módulos da imagem existente no e-mmc interno pelo novo *kernel*:

#No Snowball (Correr uma vez):

```
# Instalar ferramentas:
```

```
apt-get install rsync openssh-server
```

```
# Inserir palavra passe para acesso root:
```

```
passwd root
```

```
# montar a partição:
```

```
mkdir /boot2;mount /dev/mmcblk0p2 /boot2
```

```
# Guardar uma versão de segurança:
```

```
cp -a /boot2/uImage /boot2/uImage.org
```

#No PC:

```
# Instalar numa diretoria temporária:
```

```
mkdir modules-install
```

```
make ARCH=arm INSTALL_MOD_PATH=modules-install modules_install
```

```
# Copiar a uImage:
```

```
rsync -a arch/arm/boot/uImage root@<snowball-ip>:/boot2/uImage
```

```
# Copiar módulos:
```

```
rsync -vrl modules-install/lib/modules/3.* root@<snowball-ip>:/lib/modules
```

```
# Reiniciar:
```

```
ssh root@<snowball-ip> reboot
```

```
#Sincronizar:
```

```
sync
```

## **DESATIVAR DMA**

Existe um problema com o modo DMA no Snowball, tem de se recompilar o kernel e desativar o DMA. Para desativar, ir à diretoria fonte e editar o ficheiro .config e substituir a linha:

```
Remover: CONFIG_USB_UX500_DMA=y  
Substituir por: CONFIG_MUSB_PIO_ONLY
```

# Apêndice B: Configuração do Motion

Editar o ficheiro /etc/motion/motion.conf e seleccionar as seguintes opções:

```
#####  
# Capture device options  
#####  
# Videodevice to be used for capturing (default /dev/video0)  
  
videodevice /dev/video0  
  
# Image width (pixels). Valid range: Camera dependent, default: 352  
width 320  
  
# Image height (pixels). Valid range: Camera dependent, default: 288  
height 240  
  
# Maximum number of frames to be captured per second.  
# Valid range: 2-100. Default: 100 (almost no limit).  
framerate 20  
  
# Bitrate to be used by the ffmpeg encoder (default: 400000)  
# This option is ignored if ffmpeg_variable_bitrate is not 0 (disabled)  
ffmpeg_bps 500000  
  
# Enables and defines variable bitrate for the ffmpeg encoder.  
# ffmpeg_bps is ignored if variable bitrate is enabled.  
# Valid values: 0 (default) = fixed bitrate defined by ffmpeg_bps,  
# or the range 2 - 31 where 2 means best quality and 31 is worst.  
ffmpeg_variable_bitrate 2  
  
# Codec to used by ffmpeg for the video compression.  
# Timelapse mpegs are always made in mpeg1 format independent from this option.  
# Supported formats are: mpeg1 (ffmpeg-0.4.8 only), mpeg4 (default), and msmpeg4.
```

```

# mpeg1 - gives you files with extension .mpg
# mpeg4 or msmpeg4 - gives you files with extension .avi
# msmpeg4 is recommended for use with Windows Media Player because
# it requires no installation of codec on the Windows client.
# swf - gives you a flash film with extension .swf
# flv - gives you a flash video with extension .flv
# ffv1 - FF video codec 1 for Lossless Encoding ( experimental )
# mov - QuickTime ( testing )
ffmpeg_video_codec swf

#####
# Live Webcam Server
#####

# The mini-http server listens to this port for requests (default: 0 = disabled)
webcam_port 9090

# Quality of the jpeg images produced (default: 50)
webcam_quality 25

# Output frames at 1 fps when no motion is detected and increase to the
# rate given by webcam_maxrate when motion is detected (default: off)
webcam_motion off

# Maximum framerate for webcam streams (default: 1)
webcam_maxrate 20

# Restrict webcam connections to localhost only (default: on)
webcam_localhost off

# Limits the number of images per connection (default: 0 = unlimited)
# Number can be defined by multiplying actual webcam rate by desired number of seconds
# Actual webcam rate is the smallest of the numbers framerate and webcam_maxrate
webcam_limit 0

```

# ANEXO A

## Datasheet do multiplexer utilizado:



## Low Voltage, 300 MHz Quad 2:1 Mux Analog HDTV Audio/Video Switch

### ADG794

#### FEATURES

Bandwidth: 300 MHz

Low insertion loss and on resistance: 5  $\Omega$  typical

On-resistance flatness: 0.7  $\Omega$  typical

Single 3.3 V/5 V supply operation

Low quiescent supply current: 1 nA typical

Fast switching times

$t_{ON}$ , 7 ns

$t_{OFF}$ , 5 ns

TTL/CMOS compatible

ESD protection

2 kV human body model (HBM)

200 V machine model (MM)

1 kV field-induced charged device model (FICDM)

#### APPLICATIONS

RGB switches

HDTV

DVD-R

Audio/video switches

#### GENERAL DESCRIPTION

The ADG794 is a monolithic CMOS device comprising four 2:1 multiplexers/demultiplexers with high impedance outputs. The CMOS process provides low power dissipation yet gives high switching speed and low on resistance. The on-resistance variation is less than 1.2  $\Omega$  over the input signal range.

The wide bandwidth of the ADG794 (300 MHz typical), coupled with low distortion (0.18% typical), makes the part suitable for switching analog audio/video signals.

The ADG794 operates from a single 3.3 V/5 V supply and is TTL logic compatible. The switches are controlled by the logic inputs IN and  $\overline{EN}$ , as shown in Table 4. The  $\overline{EN}$  pin allows the user to disable all switches.

These switches conduct equally well in both directions when on. In the off condition, signal levels up to the supplies are blocked. The ADG794 switches exhibit break-before-make switching action.

The ADG794 is available in a 16-lead QSOP.

#### FUNCTIONAL BLOCK DIAGRAM

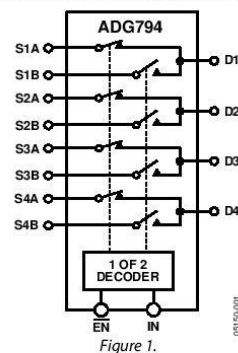


Figure 1.

#### PRODUCT HIGHLIGHTS

1. Wide bandwidth: 300 MHz.
2. Ultralow power dissipation.
3. Crosstalk: -70 dB (typical) at 10 MHz.
4. Off isolation: -65 dB (typical) at 10 MHz.
5. ESD protection tested as per ESD Association Standards:
  - 2 kV HBM (ANSI/ESD STM5.1-2001)
  - 200 V MM (ANSI/ESD STM5.2-1999)
  - 1 kV FICDM (ANSI/ESD STM5.3.1-1999)

#### Rev. B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
Tel: 781.329.4700 [www.analog.com](http://www.analog.com)  
Fax: 781.461.3113 ©2004–2008 Analog Devices, Inc. All rights reserved.

# ADG794

## TABLE OF CONTENTS

Features .....	1
Applications .....	1
Functional Block Diagram .....	1
General Description .....	1
Product Highlights .....	1
Revision History .....	2
Specifications .....	3
Single Supply .....	3
Absolute Maximum Ratings .....	5

### REVISION HISTORY

#### 2/08—Rev A to Rev B

Changes to Absolute Maximum Ratings Section, Table 3 .....	5
Updated Outline Dimensions .....	12
Changes to Ordering Guide .....	12

ESD Caution .....	5
Pin Configuration and Function Descriptions .....	6
Terminology .....	7
Typical Performance Characteristics .....	8
Typical Application .....	9
Test Circuits .....	10
Outline Dimensions .....	12
Ordering Guide .....	12

#### 4/06—Rev. 0 to Rev. A

Changes to Features Section .....	1
Changes to Product Highlights Section .....	1
Changes to Specifications Section .....	3
Changes to Typical Performance Characteristics .....	8

#### 10/04—Revision 0: Initial Version

## SPECIFICATIONS

### SINGLE SUPPLY

$V_{DD} = 5\text{ V} \pm 10\%$ ,  $GND = 0\text{ V}$ . All specifications  $T_{MIN}$  to  $T_{MAX}$ , unless otherwise noted.

Table 1.

Parameter	B Version <sup>1</sup>		Unit	Test Conditions/Comments	
	25°C	$T_{MIN}$ to $T_{MAX}$			
<b>ANALOG SWITCH</b>					
Analog Signal Range		0 to 2.5	V		
On Resistance, $R_{ON}$	5		$\Omega$ typ	$V_D = 0\text{ V}$ to $1\text{ V}$ ; $I_S = -10\text{ mA}$ ; Figure 8	
	7	8	$\Omega$ max		
On-Resistance Match Between Channels, $\Delta R_{ON}$	0.4		$\Omega$ typ	$V_D = 0\text{ V}$ to $1\text{ V}$ ; $I_S = -10\text{ mA}$	
		1.2	$\Omega$ max		
On-Resistance Flatness, $R_{FLAT(ON)}$	0.7		$\Omega$ typ	$V_D = 0\text{ V}$ to $1\text{ V}$ ; $I_S = -10\text{ mA}$	
		1.35	$\Omega$ max		
<b>LEAKAGE CURRENTS</b>					
Source Off Leakage, $I_S$ (Off)	$\pm 0.001$		nA typ	$V_S = 3\text{ V}/1\text{ V}$ ; $V_D = 1\text{ V}/3\text{ V}$ ; Figure 9	
Drain Off Leakage, $I_D$ (Off)	$\pm 0.001$		nA typ		$V_S = 3\text{ V}/1\text{ V}$ ; $V_D = 1\text{ V}/3\text{ V}$ ; Figure 9
Channel On Leakage, $I_b$ , $I_S$ (On)	$\pm 0.001$		nA typ		
<b>DIGITAL INPUTS</b>					
Input High Voltage, $V_{INH}$		2.0	V min	$V_{IN} = V_{INL}$ or $V_{INH}$	
Input Low Voltage, $V_{INL}$		0.8	V max		
Input Current					
$I_{INL}$ or $I_{INH}$	0.001		$\mu\text{A}$ typ		
		$\pm 0.1$	$\mu\text{A}$ max		
Digital Input Capacitance, $C_{IN}$		3	pF typ		
<b>DYNAMIC CHARACTERISTICS<sup>2</sup></b>					
$t_{ON}$ , $t_{ON}(\overline{EN})$	7		ns typ	$C_L = 35\text{ pF}$ ; $R_L = 50\ \Omega$ $V_S = 2\text{ V}$ ; Figure 11	
		14	ns max		
$t_{OFF}$ , $t_{OFF}(\overline{EN})$	5		ns typ	$C_L = 35\text{ pF}$ ; $R_L = 50\ \Omega$ $V_S = 2\text{ V}$ ; Figure 11	
		8	ns max		
Break-Before-Make Time Delay, $t_D$	3		ns typ	$C_L = 35\text{ pF}$ ; $R_L = 50\ \Omega$ $V_{S1} = V_{S2} = 2\text{ V}$ ; Figure 12	
		1	ns min		
Off Isolation	-65		dB typ	$f = 10\text{ MHz}$ ; $R_L = 50\ \Omega$ ; Figure 14	
Channel-to-Channel Crosstalk	-70		dB typ		$f = 10\text{ MHz}$ ; $R_L = 50\ \Omega$ ; Figure 15
Bandwidth -3 dB	300		MHz typ	$R_L = 50\ \Omega$ ; Figure 13	
THD + N	0.18		% typ	$R_L = 100\ \Omega$	
Charge Injection	7.5		pC typ	$C_L = 1\text{ nF}$ ; $V_S = 0\text{ V}$ ; Figure 16	
$C_S$ (Off)	8		pF typ		
$C_D$ (Off)	14		pF typ		
$C_D$ , $C_S$ (On)	23		pF typ		
<b>POWER REQUIREMENTS</b>					
$I_{DD}$	0.001		$\mu\text{A}$ typ	$V_{DD} = 5.5\text{ V}$ ; digital inputs = $0\text{ V}$ or $V_{DD}$	
		1	$\mu\text{A}$ max		

<sup>1</sup> Temperature range for B version is  $-40^\circ\text{C}$  to  $+85^\circ\text{C}$ .

<sup>2</sup> Guaranteed by design, not subject to production test.

# ADG794

$V_{DD} = 3\text{ V} \pm 10\%$ ,  $GND = 0\text{ V}$ . All specifications  $T_{MIN}$  to  $T_{MAX}$ , unless otherwise noted.

**Table 2.**

Parameter	B Version <sup>1</sup>		Unit	Test Conditions/Comments
	25°C	$T_{MIN}$ to $T_{MAX}$		
<b>ANALOG SWITCH</b>				
Analog Signal Range		0 to 1.5	V	
On Resistance, $R_{ON}$	7		$\Omega$ typ	$V_D = 0\text{ V}$ to $1\text{ V}$ ; $I_S = -10\text{ mA}$ ; Figure 8
	9.5	11	$\Omega$ max	
On-Resistance Match between Channels, $\Delta R_{ON}$	0.3		$\Omega$ typ	$V_D = 0\text{ V}$ to $1\text{ V}$ ; $I_S = -10\text{ mA}$
		0.9	$\Omega$ max	
On-Resistance Flatness, $R_{FLAT(ON)}$	2.6		$\Omega$ typ	$V_D = 0\text{ V}$ to $1\text{ V}$ ; $I_S = -10\text{ mA}$
		5	$\Omega$ max	
<b>LEAKAGE CURRENTS</b>				
Source Off Leakage, $I_S$ (Off)	$\pm 0.001$		nA typ	$V_S = 2\text{ V}/1\text{ V}$ ; $V_D = 1\text{ V}/2\text{ V}$ ; Figure 9
Drain Off Leakage, $I_D$ (Off)	$\pm 0.001$		nA typ	$V_S = 2\text{ V}/1\text{ V}$ ; $V_D = 1\text{ V}/2\text{ V}$ ; Figure 9
Channel On Leakage, $I_D$ , $I_S$ (On)	$\pm 0.001$		nA typ	$V_D = V_S = 2\text{ V}/1\text{ V}$ ; Figure 10
<b>DIGITAL INPUTS</b>				
Input High Voltage, $V_{INH}$		2.0	V min	
Input Low Voltage, $V_{INL}$		0.8	V max	
Input Current				
$I_{INL}$ or $I_{INH}$	0.001		$\mu\text{A}$ typ	$V_{IN} = V_{INL}$ or $V_{INH}$
		$\pm 0.1$	$\mu\text{A}$ max	
Digital Input Capacitance, $C_{IN}$		3	pF typ	
<b>DYNAMIC CHARACTERISTICS<sup>2</sup></b>				
$t_{ON}$ , $t_{ON}(\overline{EN})$	10		ns typ	$C_L = 35\text{ pF}$ ; $R_L = 50\ \Omega$
		16	ns max	
$t_{OFF}$ , $t_{OFF}(\overline{EN})$	6		ns typ	$V_S = 1.5\text{ V}$ ; Figure 11
		10	ns max	$C_L = 35\text{ pF}$ ; $R_L = 50\ \Omega$
Break-Before-Make Time Delay, $t_D$	3		ns typ	$V_S = 1.5\text{ V}$ ; Figure 11
		1	ns min	$C_L = 35\text{ pF}$ ; $R_L = 50\ \Omega$
Off Isolation	-65		dB typ	$V_{S1} = V_{S2} = 1.5\text{ V}$ ; Figure 12
Channel-to-Channel Crosstalk	-70		dB typ	$f = 10\text{ MHz}$ ; $R_L = 50\ \Omega$ ; Figure 14
Bandwidth -3 dB	300		MHz typ	$f = 10\text{ MHz}$ ; $R_L = 50\ \Omega$ ; Figure 15
THD + N	0.18		% typ	$R_L = 50\ \Omega$ ; Figure 13
Charge Injection	4		pC typ	$R_L = 100\ \Omega$
$C_S$ (Off)	8		pF typ	$C_L = 1\text{ nF}$ ; $V_S = 0\text{ V}$ ; Figure 16
$C_D$ (Off)	14		pF typ	
$C_D$ , $C_S$ (On)	23		pF typ	
<b>POWER REQUIREMENTS</b>				
$I_{DD}$	0.001		$\mu\text{A}$ typ	$V_{DD} = 3.3\text{ V}$ ; digital inputs = $0\text{ V}$ or $V_{DD}$
		1	$\mu\text{A}$ max	

<sup>1</sup> Temperature range for B version is  $-40^\circ\text{C}$  to  $+85^\circ\text{C}$ .

<sup>2</sup> Guaranteed by design, not subject to production test.

## ABSOLUTE MAXIMUM RATINGS

$T_A = 25^\circ\text{C}$ , unless otherwise noted.

Table 3.

Parameters	Ratings
$V_{DD}$ to GND	-0.3 V to +6 V
Analog, Digital Inputs <sup>1</sup>	-0.3 V to $V_{DD} + 0.3$ V or 30 mA, whichever occurs first
Continuous Current, S or D	100 mA
Peak Current, S or D	300 mA (pulsed at 1 ms, 10% duty cycle maximum)
Operating Temperature Range Industrial (B Version)	-40°C to +85°C
Storage Temperature Range	-65°C to +150°C
Junction Temperature	150°C
QSOP Package, Power Dissipation	566 mW
$\theta_{JA}$ Thermal Impedance	149.97°C/W
Lead Temperature, Soldering Reflow, Peak Temperature	260(+0/-5)°C
Time at Peak Temperature	20 sec to 40 sec

<sup>1</sup> Overvoltages at IN, S, or D are clamped by internal diodes. Current should be limited to the maximum ratings given.

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Only one absolute maximum rating may be applied at any one time.

Table 4. Truth Table

EN	IN	D1	D2	D3	D4	Function
1	X	High-Z	High-Z	High-Z	High-Z	Disable
0	0	S1A	S2A	S3A	S4A	IN = 0
0	1	S1B	S2B	S3B	S4B	IN = 1

### ESD CAUTION



**ESD (electrostatic discharge) sensitive device.** Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

# ADG794

## PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

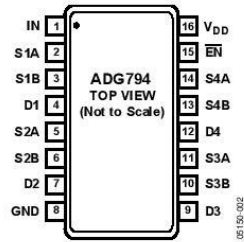


Figure 2. Pin Configuration

Table 5. Pin Function Descriptions

Pin NO.	Mnemonic	Description
1	IN	Logic Control Input. The logic level at this input controls the operation of the multiplexers (see Table 4).
2	S1A	A-Side Source Terminal of Mux1. Can be an input or an output.
3	S1B	B-Side Source Terminal of Mux1. Can be an input or an output.
4	D1	Drain Terminal of Mux1. Can be an input or an output.
5	S2A	A-Side Source Terminal of Mux2. Can be an input or an output.
6	S2B	B-Side Source Terminal of Mux2. Can be an input or an output.
7	D2	Drain Terminal of Mux2. Can be an input or an output.
8	GND	Ground Reference.
9	D3	Drain Terminal of Mux3. Can be an input or an output.
10	S3B	B-Side Source Terminal of Mux3. Can be an input or an output.
11	S3A	A-Side Source Terminal of Mux3. Can be an input or an output.
12	D4	Drain Terminal of Mux4. Can be an input or an output.
13	S4B	B-Side Source Terminal of Mux4. Can be an input or an output.
14	S4A	A-Side Source Terminal of Mux4. Can be an input or an output.
15	EN	Mux Enable Logic Input. Enables or disables the multiplexers (see Table 4).
16	V <sub>DD</sub>	Positive Power Supply Voltage.

## TERMINOLOGY

### $V_{DD}$

Most positive power supply potential.

### $I_{DD}$

Positive supply current.

### GND

Ground (0 V) reference.

### S

Source terminal. Can be either an input or an output.

### D

Drain terminal. Can be either an input or an output.

### IN

Logic control input.

### $V_D$ ( $V_S$ )

Analog voltage on Terminal D and Terminal S.

### $R_{ON}$

Ohmic resistance between Terminal D and Terminal S.

### $R_{FLAT (ON)}$

Flatness is defined as the difference between the maximum and minimum value of on resistance as measured.

### $\Delta R_{ON}$

On-resistance match between any two channels.

### $I_S$ (Off)

Source leakage current with the switch off.

### $I_D$ (Off)

Drain leakage current with the switch off.

### $I_D, I_S$ (On)

Channel leakage current with the switch on.

### $V_{INL}$

Maximum input voltage for Logic 0.

### $V_{INH}$

Minimum input voltage for Logic 1.

### $I_{INL}$ ( $I_{INH}$ )

Input current of the digital input.

### $C_S$ (Off)

Off switch source capacitance. Measured with reference to ground.

### $C_D$ (Off)

Off switch drain capacitance. Measured with reference to ground.

### $C_D, C_S$ (On)

On switch capacitance. Measured with reference to ground.

### $C_{IN}$

Digital input capacitance.

### $t_{ON}$

Delay time between the 50% and the 90% points of the digital input and switch on condition.

### $t_{OFF}$

Delay time between the 50% and the 90% points of the digital input and switch off condition.

### $t_{BBM}$

On or off time measured between the 80% points of both switches when switching from one to another.

### Charge Injection

A measure of the glitch impulse transferred from the digital input to the analog output during on/off switching.

### Off Isolation

A measure of unwanted signal coupling through an off switch.

### Crosstalk

A measure of unwanted signal that is coupled through from one channel to another as a result of parasitic capacitance.

### -3 dB Bandwidth

The frequency at which the output is attenuated by 3 dB.

### On Response

The frequency response of the on switch.

### Insertion Loss

The loss due to the on resistance of the switch.

### THD + N

The ratio of the harmonic amplitudes plus the noise of a signal to the fundamental.

TYPICAL PERFORMANCE CHARACTERISTICS

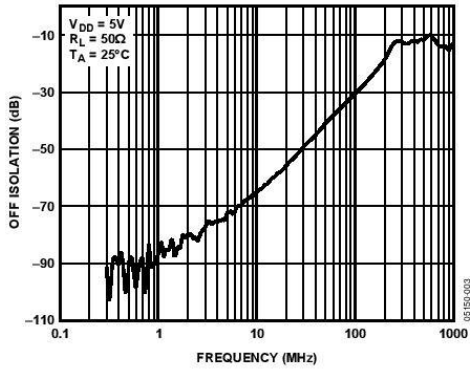


Figure 3. Off Isolation vs. Frequency

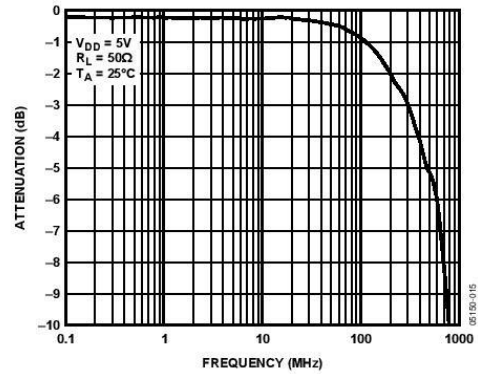


Figure 5. Bandwidth

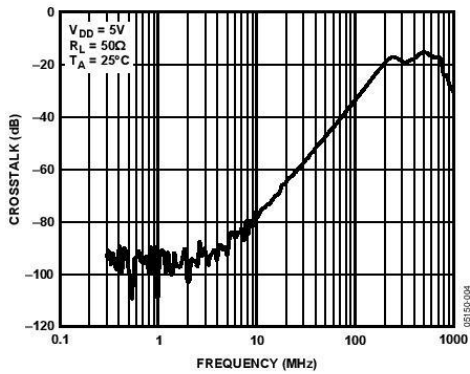


Figure 4. Crosstalk vs. Frequency

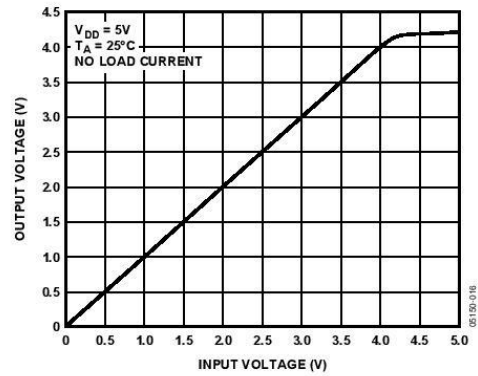


Figure 6. Output Voltage vs. Input Voltage

TYPICAL APPLICATION

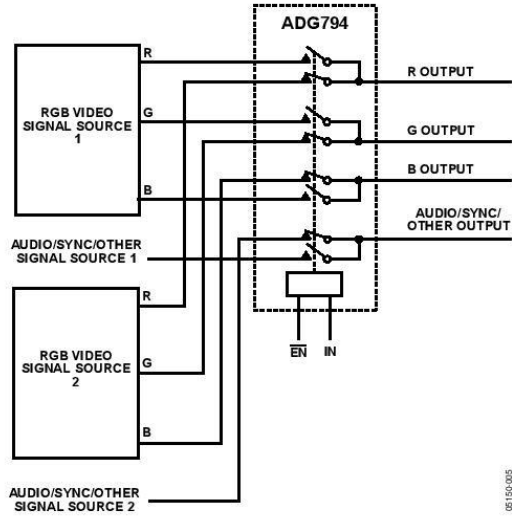


Figure 7. Audio/Video Switch

06159-005

# ADG794

## TEST CIRCUITS

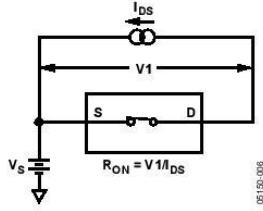


Figure 8. On Resistance

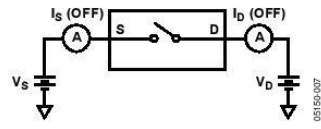


Figure 9. Off Leakage

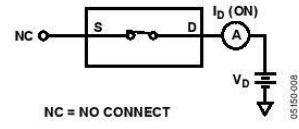


Figure 10. On Leakage

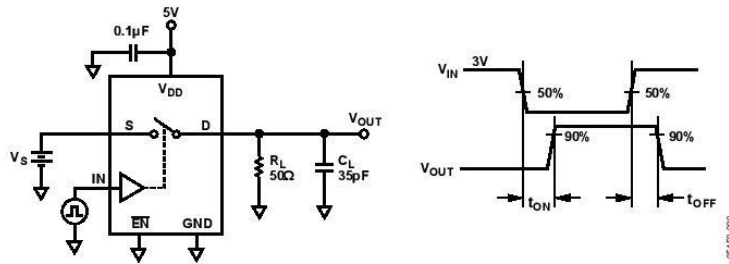


Figure 11. Switching Times

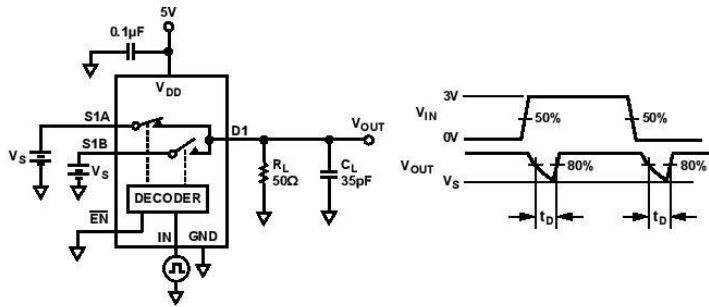


Figure 12. Break-Before-Make Time Delay

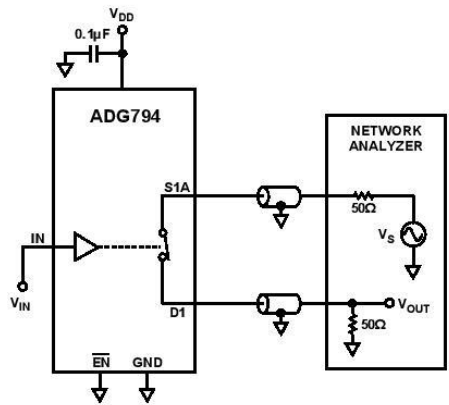


Figure 13. Bandwidth

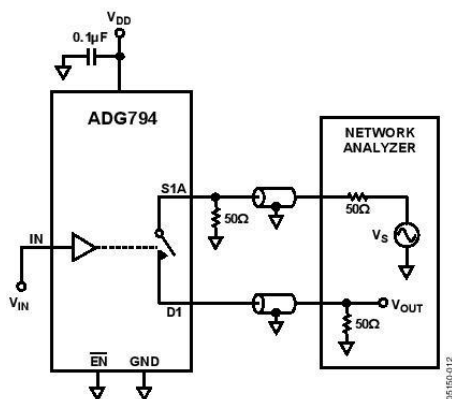


Figure 14. Off Isolation

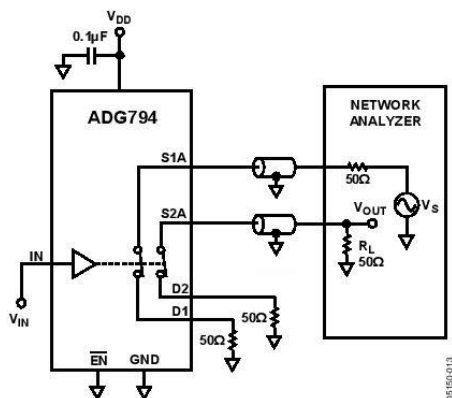


Figure 15. Channel-to-Channel Crosstalk

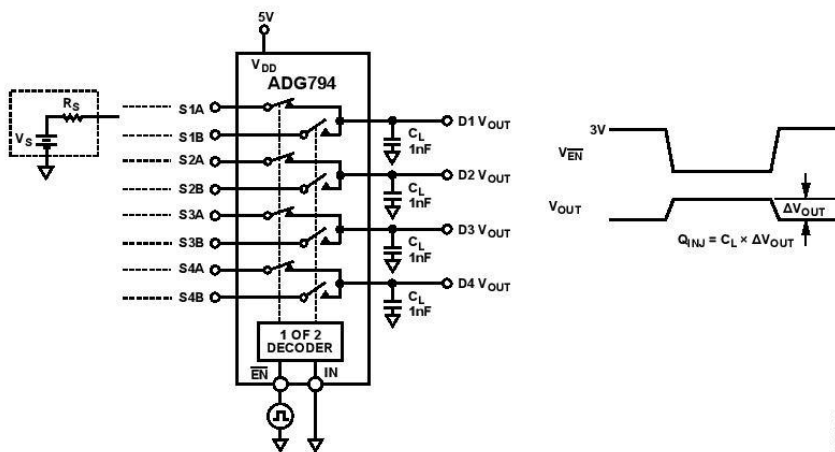
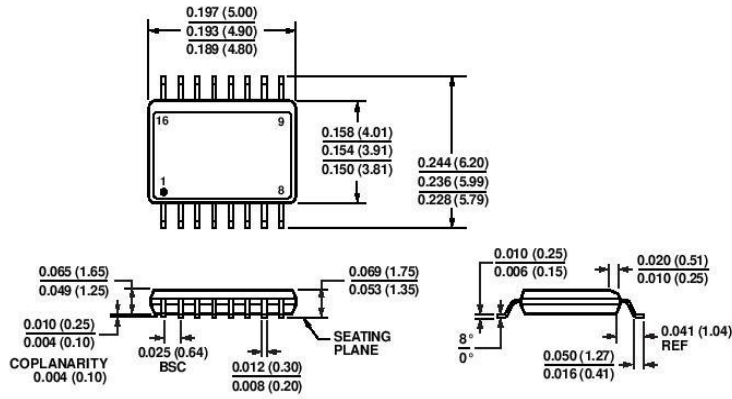


Figure 16. Charge Injection

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-137-AB  
 CONTROLLING DIMENSIONS ARE IN INCHES; MILLIMETERS DIMENSIONS  
 (IN PARENTHESES) ARE ROUNDED-OFF INCH EQUIVALENTS FOR  
 REFERENCE ONLY AND ARE NOT APPROPRIATE FOR USE IN DESIGN.

Figure 17. 16-Lead Shrink Small Outline Package [QSOP]  
 (RQ-16)

Dimensions shown in inches and (millimeters)

01286E-A

ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option
ADG794BRQZ <sup>1</sup>	-40°C to +85°C	16-Lead Shrink Small Outline Package (QSOP)	RQ-16
ADG794BRQZ-500RL7 <sup>1</sup>	-40°C to +85°C	16-Lead Shrink Small Outline Package (QSOP)	RQ-16
ADG794BRQZ-REEL <sup>1</sup>	-40°C to +85°C	16-Lead Shrink Small Outline Package (QSOP)	RQ-16
ADG794BRQZ-REEL7 <sup>1</sup>	-40°C to +85°C	16-Lead Shrink Small Outline Package (QSOP)	RQ-16

<sup>1</sup> Z = RoHS Compliant Part.

# ANEXO B

## Especificações do Snowball

Version	SDK
Processor	A9500
PoP Memory	8Gb Micron LPDDR2
eMMC	4GBytes
PMIC	AB8500
Debug Support	HE10-2x10 header for JTAG and MIPI Debug Connector
Console	Serial over USB (J21)
HS USB OTG	Mini-AB USB connector
Audio	Audio out/in on 3.5mm jack connector
Memory card	microSD socket
Video Output	HDMI™ connector, CVBS
Connectivity	10/100 Ethernet and WIFI Bluetooth® GPS Combo Module
Sensors	3D Accelerometer/Magnetometer, 3D Gyrometer and Pressure sensor
Power connector	USB OTG or 2.0mm jack connector or Li-ion battery
Push buttons	Reset and User push buttons
Indicators	1x User LED + 1 Power LED + 1 Charger LED
Expansion connector	1x HE13-1x16

# ANEXO C

## Características da webcam

Carl Zeiss® lens

Autofocus system

Ultra-high resolution 2-megapixel sensor with RightLight™, 2 Technology

Color depth: 24-bit true color

Video capture: Up to 1600 x 1200 pixels (HD Video 960 x 720 pixels)

Frame rate: Up to 30 frames per second

Still image capture: 8 million pixels (with software enhancement)

Built-in microphone with RightSound Technology

## 9. Bibliografia

1. **Tesla, Nikola.** Tesla Memorial Society of New York. *Tesla Radio*. [Online] [Cited: 08 22, 2012.] <http://www.teslasociety.com/radio.htm>.
2. **Taylor, John W. R. and Munson, Kenneth.** *Jane's pocket book of remotely piloted vehicles*. s.l. : Colier Books, 1977. 002080640X.
3. Barnard Microsystems Limited. [Online] [Cited: 10 30, 2012.] [http://www.barnardmicrosystems.com/L4E\\_automation.htm](http://www.barnardmicrosystems.com/L4E_automation.htm).
4. **Pashtan, Ariel.** *Wireless Terrestrial Communications: Cellular Telephony*. s.l. : Aware Networks, inc., 2006.
5. *Father of the cell phone*. **The Economist**. 2009.
6. **AT&T.** *1946: First Mobile Telephone Call*. [Online] [Cited: 10 30, 2012.] <http://www.corp.att.com/atlabs/reputation/timeline/46mobile.html>.
7. **Stojmenovic, Ivan.** *HANDBOOK OF WIRELES NETWORKS AND MOBILE COMPUTING*. s.l. : John Wiley & Sons, Inc., 2002. 0-471-22456-1.
8. Telecom ABC. *GPRS*. [Online] [Cited: 10 30, 2012.] <http://www.telecomabc.com/g/gprs.html>.
9. EVERYMAC.com. [Online] [Cited: 10 30, 2012.] <http://www.everymac.com/systems/apple/iphone/iphone-faq/iphone-edge-3g-support-difference-between-edge-and-3g-mvno.html>.
10. **Oliveira, Anderson Clayton.** Teleco. *Tutoriais*. [Online] [Cited: 10 30, 2012.] <http://www.teleco.com.br/tutoriais.asp>.
11. IMT-Advanced. *www.telecomabc.com*. [Online] [Cited: 10 30, 2012.] <http://www.telecomabc.com/i/imt-advanced.html>.
12. **Calao Systems.** *Hardware Reference Manual*. 2011.
13. **Analog Devices.** *ADG794 Datasheet*. s.l. : Analog Devices.
14. **Kurose, James F. and Ross, Keith W.** *Computer Networking: A Top-Down Approach (5th Edition)*. s.l. : Addison Wesley, 2009. 0-13-607967-9.
15. Tutorial Gratis. [Online] [Cited: 09 11, 2012.] <http://www.tutorialgratis.com.br/software/107-ddns-como-funciona-encontre-seu-computador-na-internet-ace>.
16. Java.net. [Online] [Cited: 10 30, 2012.] <http://java.net/projects/jinput>.
17. **Négrier, Aldric Trindade.** *Alternativas de controlo de veículos por rádio frequência: Uma abordagem utilizando a telerobótica via Internet através da rede de telefonia celular*. *Dissertação de Mestrado*. 2010.