



UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE
COMPUTAÇÃO



**Projeto e Implementação de um Piloto Automático Aplicado a
Aeromodelos de Asa Fixa e Asa Móvel**

RUBENS ANTÔNIO ALVES

Orientador: Prof. Dr. José Wilson Lima Nerys

Goiânia 2015

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR AS TESES E DISSERTAÇÕES ELETRÔNICAS (TEDE) NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: ☒ **Dissertação** ☐ **Tese**

2. Identificação da Tese ou Dissertação

Autor (a):		Rubens Antonio Alves	
E-mail:		rubensantonioa@gmail.com	
Seu e-mail pode ser disponibilizado na página? <input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não			
Vínculo empregatício do autor		Engenheiro Eletricista	
Agência de fomento:		CELG-D - ANEEL	Sigla: P&D
País:	Brasil	UF:	GO CNPJ: 01.543.032/0001-04
Título:	Projeto e Implementação de um Piloto Automático Aplicado a Aeromodelos de Asa Fixa e asa Móvel.		
Palavras-chave:	Piloto Automático, Controlador, Asa Fixa. Asa Rotativa. Voo Autônomo. Voo Assistido.		
Título em outra língua:	Design and Implementation of a Autopilot Aplied a Model airplane.		
Palavras-chave em outra língua:	Automatic Pilot. Controller. Fixed-Wing Drone. Rotary-wing. GPS guided Flight. Manual Flight.		
Área de concentração:	Engenharia Elétrica		
Data defesa: (dd/mm/aaaa)	30/09/2015		
Programa de Pós-Graduação:	Programa de Pós-graduação em Engenharia Elétrica e de Computação		
Orientador (a):	José Wilson Lima Nerys		
E-mail:	jwlnerys@gmail.com		
Co-orientador (a):*			
E-mail:			

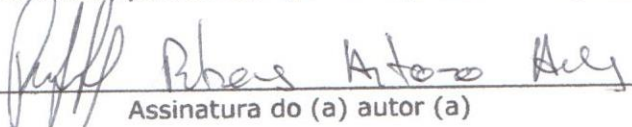
*Necessita do CPF quando não constar no SisPG

3. Informações de acesso ao documento:

Concorda com a liberação total do documento ☒ **SIM** ☐ **NÃO¹**

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF ou DOC da tese ou dissertação.

O sistema da Biblioteca Digital de Teses e Dissertações garante aos autores, que os arquivos contendo eletronicamente as teses e ou dissertações, antes de sua disponibilização, receberão procedimentos de segurança, criptografia (para não permitir cópia e extração de conteúdo, permitindo apenas impressão fraca) usando o padrão do Acrobat.


 Assinatura do (a) autor (a)

Data: 04 / 11 / 2015

¹ Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

RUBENS ANTÔNIO ALVES

**PROJETO E IMPLEMENTAÇÃO DE UM PILOTO
AUTOMÁTICO APLICADO A AEROMODELOS DE ASA
FIXA E ASA MÓVEL**

Dissertação apresentada ao Curso de Mestrado
em Engenharia Elétrica e de Computação da
Universidade Federal de Goiás, para obtenção do
título de Mestre em Engenharia Elétrica.

Área de Concentração: Engenharia Elétrica.

Orientador: Prof. Dr. Jose Wilson Lima Nerys.

Goiânia

2015

Ficha catalográfica elaborada automaticamente
com os dados fornecidos pelo(a) autor(a), sob orientação do Sibi/UFG.

Alves, Rubens Antonio

Projeto e Implementação de um Piloto Automático Aplicado a
Aeromodelos de Asa Fixa e Asa Móvel [manuscrito] / Rubens Antonio
Alves. - 2015.

xvii, 116 f.: il.

Orientador: Prof. Dr. Dr. José Wilson Lima Nerys .
Dissertação (Mestrado) - Universidade Federal de Goiás, Escola de
Engenharia Elétrica (EEEC) , Programa de Pós-Graduação em
Engenharia Elétrica e de Computação, Goiânia, 2015.

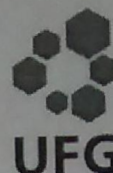
Bibliografia. Anexos.

Inclui siglas, mapas, fotografias, símbolos, tabelas, lista de figuras,
lista de tabelas.

1. Vant. 2. Piloto Automático. I. Lima Nerys , Dr. José Wilson,
orient. II. Título.



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO
COORDENAÇÃO DE PÓS-GRADUAÇÃO

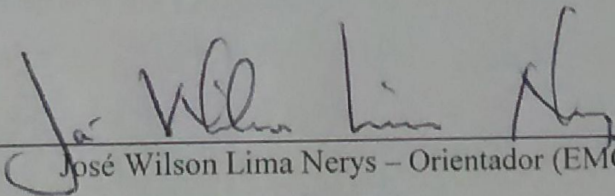


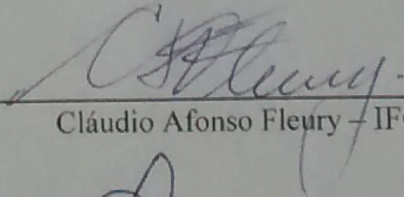
FOLHA DE APROVAÇÃO

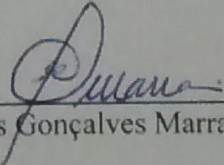
"Projeto e Implementação de um Piloto Automático Aplicado a Aeromodelos de Asa Fixa e Asa Móvel"

RUBENS ANTÔNIO ALVES

Dissertação defendida e aprovada pela banca examinadora constituída pelos senhores:


José Wilson Lima Nerys – Orientador (EMC/UFG)


Cláudio Afonso Fleury – IFG


Enes Gonçalves Marra – EMC/UFG

Goiânia, 30 de setembro de 2015

DEDICATÓRIA

Dedico este trabalho primeiramente a
Deus pelo dom da vida,
à memória de meus pais,
por sempre me apoiarem em meus estudos
e pela paciência em me ensinar,
que não existe maior riqueza
que o conhecimento
e não haverá maior herança
que a educação.

AGRADECIMENTOS

Agradeço a meus filhos por tudo o que representam na minha vida, agradeço a minha esposa por me incentivar nos estudos.

A meu orientador José Wilson Lima Nerys por todo empenho e ajuda, sem o mesmo não seria possível sequer iniciar este projeto, o qual me abriu as portas desta universidade, e me ajudou crescer não apenas na profissão, mas intelectualmente.

A meus irmãos e toda minha família, pelo apoio e incentivo aos estudos, e compreensão pela ausência em família.

A todos meus amigos de escola pelo companheirismo, que na oportunidade cito três amigos que estiveram mais presentes nesta última etapa, Marcelino, Andre e Gustavo.

A todos os professores do curso, em especial ao professor Bernardo pela matéria lecionada sobre motor brushless, muito utilizado neste projeto e que me ajudara a compreender muitas características do mesmo.

A todos meus amigos que no decorrer de minha profissão, me ajudarão a superar as pedras e obstáculos do caminho para que eu pudesse chegar aqui.

A todos que anteriormente a este trabalho ser iniciado já serviam de referência não apenas como bibliografia, mas também como inspiração que seria possível alcançar meus objetivos.

A tres amigos de longa data, Ismel Batista, Antônio Sampaio e Adilsom Sakihi que graças ao incentivo deles hoje também tenho uma profissão.

Agradeço a todos os mestres que tive durante minha vida de estudante, pessoas simples que fizeram de seu trabalho e dedicação, exemplo de um mundo de possibilidades, onde a certeza do sucesso pessoal sempre foi à superação de seus alunos, em especial a minha primeira professora ‘Tia Regina’, a qual em minha memória ainda sinto, o calor de sua mão, segurando a minha para sobreescrever o traçado das letras do alfabeto.

“e você aprende
a construir todas as estradas hoje
porque o terreno de amanhã é
demasiado incerto para planos
e futuros têm o hábito de cair
no meio do vôo”

Veronica Shoffstall

RESUMO

Esta dissertação da área de engenharia elétrica consiste na construção de um sistema de controle completo de hardware e software para controle de um aeromodelo de asa fixa e/ou asa móvel, de maneira a implementar um piloto automático compatível com as necessidades de voo autônomo ou assistido, sendo que tal sistema deverá ser compatível com a precisão de voo aplicada a monitoramento de sistemas críticos. O sistema é constituído por uma placa controladora composta por hardware e firmware específicos, capaz de controlar um modelo orientado por coordenadas GPS, para que o mesmo percorra uma rota predeterminada e retorne ao ponto de partida de forma autônoma. A placa recebe, ainda em solo, a programação da rota a ser percorrida; o aeromodelo deve responder normalmente aos comandos do controle remoto do piloto em solo, respeitando o raio de alcance visual do piloto no modo assistido e segue a rota programada no modo autônomo após confirmação de comando do piloto. No final do percurso o aeromodelo volta em trajetória reta para o ponto de partida, respeitando a maior altura do trajeto. O aeromodelo deve ser munido de sistema de comunicação para o devido acompanhamento em solo das condições de aeronavegabilidade do aeromodelo em voo, bem como a verificação dos níveis de precisão em relação à rota programada. A comunicação pode ser feita diretamente por sistema de link de rádio, direcionada a um terminal móvel em solo, que é acompanhado pelo piloto, para aumentar o nível de segurança, mas o aeromodelo também pode comunicar por rede de celular GPRS, conectado à internet, que direciona os dados ao terminal em solo. Nesse caso, o terminal, também deverá estar conectado à internet. A rota programada no aeromodelo é construída com base em mapas online conectados diretamente ao software de programação e acompanhamento de missão, que faz o tratamento e armazenamento dos dados e parâmetros do aeromodelo. Tanto a programação de controle de estabilidade, quanto de rota, com dados primários relativos a latitude, longitude e altura, permitem o acompanhamento em tempo real do aeromodelo junto à rota programada e também através da imagem da câmera de gravação embarcada no aeromodelo. Todos os dados são gravados com base em processo de linha do tempo, que podem ser recuperados em conjunto para análise posterior.

Palavras-chave: Piloto Automático, Controlador, Asa Fixa. Asa Rotativa. Voo Altonomo. Voo Assistido. Internet.

ABSTRACT

This Project, in the electrical engineering area, consists in the development of a complete control system, hardware and software, for controlling model airplanes of the fixed-wing and rotary-wing types, aiming the implementation of an automatic control system compatible with the necessity of autonomous and aided flights, applied to critical systems monitoring. The final system consists of a controller, or automatic pilot, with specific hardware and software, capable of controlling a model airplane using GPS coordinates, in a way that allows the airplane to go through a planned route and go back to the starting point in an autonomous way. The controller should receive, in ground, the programmed route; the model should answer to the pilot commands, within a visual range when operating in the aided mode, and should go through the programmed route in the autonomous mode, after confirmation of the pilot. After reaching the end of the programmed route, the model airplane should return to the starting point, keeping the maximum flight level of the route as the reference height. The model airplane will carry in a communication system to allow the monitoring process from a ground station, able to keep updated the airworthy conditions, as well as the level of accuracy between the actual and the planned route. The communication may be carried out directly using a radio link, with the receiver allocated in a mobile ground station, monitored by a pilot, to make higher the security level. However, the model airplane may transfer the data through a GPRS link, connected to the web system, which transfers the data to the ground station. In this case, the ground station must be connected to the web. The route saved in the model control system is built based on online maps directly linked by the software for the mission programming and monitoring, which can carry out the treatment and storage of the model data and parameters. The programming of the stability control and route, with primary data of latitude, longitude and height allows the real time monitoring of the model, related to the planned route and through images captured by embedded video cameras. All data are storage following a timeline process, such that they can be recovered for further analysis.

Keywords: Automatic Pilot. Controller. Fixed-Wing. Drone. Rotary-wing. GPS guided Flight. Manual Flight.

LISTA DE FIGURAS

Figura 1: Mapa de interligação dos sistemas ONS 2011	21
Figura 2: DSP Microchip 16bits (MICROCHIP T. , 2011)	26
Figura 3: Controlador de reset (MICROCHIP T. , 2011).	27
Figura 4: Pinos de conexão da porta serial (MICROCHIP T. , 2011).....	28
Figura 5: Temporizador 1 com oscilador de entrada (MICROCHIP T. , 2011).	29
Figura 6: Interface SPI (MICROCHIP T. , 2011).	30
Figura 7: Comunicação I2C com memória (MICROCHIP T. , 2011).....	31
Figura 8: Interface I2C (MICROCHIP T. , 2011).	31
Figura 9 Interface ADC (MICROCHIP T. , 2011).....	32
Figura 10: Input Capture (MICROCHIP T. , 2011).	33
Figura 11: Output Capture (MICROCHIP T. , 2011).....	34
Figura 12: Nano injetor (fonte: (Brian Jensen/BYU)) e um ácaro sobre estrutura (fonte: (SOARES DE LIMA, 2005))	36
Figura 13: Eixos de referência (Fonte (SOARES DE LIMA, 2005)).....	38
Figura 14: Fluxo de Ar Sobre a Asa (Fonte: Portal ime.unicamp).....	39
Figura 15: Forças de Atuação em Um Avião (Fonte: (FRANCISCO, 2012)).....	40
Figura 16: Estruturas Básicas de Controle um Avião (Fonte: (FRANCISCO, 2012)).....	41
Figura 17: Movimentos de Controle em Relação ao Centro de Massa – Fonte: (ARAÚJO., 2007)	42
Figura 18: Constelação de satélites (Fonte: Portal Vaz Tolentino)	44
Figura 19: Módulo GPS Ublox 6m e Atena (Fonte: Aliexpress)	46
Figura 20 Erro offset ocasionado por proximidade de campo magnético	47
Figura 21: Sensor magnetometro posicionado a 8 metros da linha de transmissão.	48
Figura 22 Medidas do sensor a 70 m e 35 m de distância da linha de distribuição.....	48
Figura 23 (a): Sensor magnético HMC5883 ao lado de trafo 1 kVA. (b): filtragem	49
Figura 24 Conexão da nova placa de sensores ao barramento I2C	50
Figura 25: Sensor Giroscópio e Acelerômetro (fonte: (INVENCENCE, 2013)).....	51
Figura 26 Diagrama de blocos do circuito.	53
Figura 27 Esboço do circuito da placa controladora V1.0.....	54
Figura 28 layout 3D superior e inferior V1.0.....	55
Figura 29 Placa já confeccionada V1.0.....	56

Figura 30 Placa montada.....	56
Figura 31 Mapa de memória do DSPIC33EP512 (MICROCHIP T. , 2011).....	58
Figura 32 Fluxograma do software.....	59
Figura 33 Sinal Serial PPM (Fonte: Portal Futaba).....	61
Figura 34 Sinal analógico para controle de servo. (Fonte Junior, Siqueira).....	62
Figura 35 Modelo de controle utilizado.....	66
Figura 36 Curva glissada utilizando ailerons e forças resultantes (fonte (ARAUJO., 2007)).	67
Figura 37 Controlador aplicado ao modelo de asas fixa.....	68
Figura 38 Tela de configuração de parâmetros do modelo.....	70
Figura 39: Imagem da câmera de bordo e acompanhamento de rota no SPAM.....	71
Figura 40 Janela de comunicação entre o controle do aeromodelo e a placa controladora	73
Figura 41 Sinal de controle (modulado e decodificado) recebido pela placa controladora.....	73
Figura 42 Acionamento de 4 servo-motores usando a placa controladora.....	74
Figura 43 - Posição do <i>stick</i> e do servo (a 0 grau) e sinais modulado (modulação PPM) e decodificado	74
Figura 44-Posição do stick e do servo (a 60 graus) e sinais modulado (modulação PPM) e decodificado	75
Figura 45 Arquivo CELG.TXT de armazenamento dos dados de voo.....	75
Figura 46 Tela de Seleção do modelo no simulador FlightGear.....	77
Figura 47 Configuração do aeroporto da simulação.....	78
Figura 48 Configuração da interface de saída (FLIGHTGEAR, 2015).....	79
Figura 49 Cabo adaptador PPM/USB.....	80
Figura 50 Fluxo de sinais para simulação da placa	81
Figura 51 Atuação da placa controladora em voo nivelado no simulador V1.9.1. (FLIGHTGEAR, 2015).....	83
Figura 52 Rota programada e percorrida no simulador.....	84
Figura 53 Teste de estabilização.....	86
Figura 54 Rota com dois pontos.....	87
Figura 55 rota percorrida em modo autônomo.....	88
Figura 56 Pontos de recálculo da rota.....	89
Figura 57 Subpontos da rota para recálculo.....	90
Figura 58 Palca com uso do recálculo de rota com subpontos.....	91
Figura 59 Vista 3D da placa versão 1.2	102

Figura 60 Placa montada versão 1.2 <i>layout</i> superior	102
Figura 61 Placa montada versão 1.2 <i>layout</i> inferior	103
Figura 62 Módulo GPRS.....	104
Figura 63 Conexões do DSP.	105
Figura 64 Leds de indicação e sensores.	106
Figura 65 Conexões de servos e portas seriais.	107
Figura 66 Fonte de alimentação.....	108

LISTA DE TABELAS

Tabela 1: Extensão das Linhas de Transmissão do SIN – km.....	20
Tabela 2 Exemplo de pacote enviado ao modelo.	64
Tabela 3 Serviços implementados na versão 1.0.5.1 SPAM.....	65

LISTA DE ABREVIACÕES

A/D	Analógico/Digital
AC	Alternating current (corrente alternada)
ASCII	<i>American Standard Code for Information Interchange</i> (código padrão americano para o intercâmbio de códigos)
CI	Circuito integrado
CPU	<i>Central Processing Unit</i> (unidade central de processamento)
D/A	Digital/Analógico
DC	<i>Direct current</i> (corrente contínua)
DMP	Digital Motion Processor (processador de movimentos)
TSSOP	<i>Thin Shrink Small-Outline Package</i> (encapsulamento para montagem de superfície)
EEPROM	<i>Erasable Programmable Read-Only Memory</i> (memória programável só de leitura apagável)
FIFO	<i>First In, First Out</i> (primeiro a entrar, primeiro a sair)
GPS	<i>Global Positioning System</i> (Geo-Posicionamento por Satélite)
GPRS	<i>General Packet Radio Services</i> (Serviços Gerais de Pacote por Rádio)
I2C	<i>Inter-Integrated Circuit</i> (Circuito Inter-interintegrado)
IP	<i>Internet Protocol</i> (protocolo de internet)
MMC	<i>MultiMedia Card</i> (Cartão Multimidia)
PCB	<i>Printed circuit board</i> (placa de circuito impresso)
PPM	<i>Pulse-position modulation</i> (Modulação por posição de pulso)
PROM	<i>Programmable Read-Only Memory</i> (memória programável só de leitura)
PWM	<i>Pulse Width Modulation</i> (modulação por largura de pulso)
RAM	<i>Random Access Memory</i> (memória de acesso aleatório)
ROM	<i>Read Only Memory</i> (memória só de leitura)
SMD	<i>Surface mounted device</i> (dispositivos de soldagem superficial)
SPI	<i>Serial Peripheral Interface</i> (interface serial de periféricos)
UART	<i>Universal Asynchronous Receiver/Transmitter</i> (transmissor/receptor universal assíncrono)

VANT Veículo Aéreo Não Tripulado.

LISTA DE SÍMBOLOS

Vcc	Valor de tensão positiva digital
GND	Valor de tensão de referência digital
VDD	Valor de tensão positiva digital
VSS	Valor de tensão de referência digital
AVDD	Valor de tensão positiva analógica
AVSS	Valor de tensão de referencia analógica
VREF+	Valor de tensão positiva analógica de referência
VREF-	Valor de tensão negativa analógica de referência

SUMÁRIO

1 INTRODUÇÃO.....	19
1.1 OBJETIVOS.....	19
1.2 MOTIVAÇÃO.....	20
1.3 ORGANIZAÇÃO DO TEXTO	22
2 REVISÃO BIBLIOGRÁFICA	23
2.1 HISTÓRICO.....	23
2.2 MICROCONTROLADORES E APLICAÇÕES EM VANTS/SISTEMA EMBARCADOS.....	24
2.3 PROCESSADORES DIGITAIS DE SINAIS (DSP).....	24
2.3.1 Aplicações de Microcontroladores e DSPs	25
2.3.2 DSPIC.....	26
2.3.2.1 Reset	27
2.3.2.2 (UART)– Universal Asynchronous Receiver Transmitter.....	27
2.3.2.3 (TIMER)- Temporizador	28
2.3.2.4 Serial Peripheral Interface (SPI)	29
2.3.2.5 Inter-Integrated Circuit.....	30
2.3.2.6 Conversor Analógico-Digital (ADC)	32
2.3.2.7 Peripheral Pin Select (PPS).....	34
2.3.2.8 Input Capture (IC).....	33
2.3.2.9 Output Capture (OC).....	34
2.4 SISTEMAS MICRO-ELETROME CÂNICOS (MEMS).....	35
3 PRINCÍPIOS AERODINÂMICOS DE UM AEROMODELO.	37
3.1 SISTEMAS DE REFERÊNCIA PARA UM CORPO NO ESPAÇO.....	37
3.2 DINÂMICA PARA MODELO DE ASA FIXA.....	38
3.2.1 Estruturas e Superfícies de Controle de um avião	40
3.2.2 Movimentos de controle de um Avião	42
4 PROJETO E IMPLEMENTAÇÃO DA PLACA CONTROLADORA.....	44
4.1 SENSORES	44
4.1.1 GPS.....	44
4.1.2 Magnetômetro.....	46
4.1.3 Giroscópio Acelerômetro	51
4.2 CIRCUITO ELÉTRICO DA PLACA.....	52
4.3 PLACA CONTROLADORA EM CIRCUITO IMPRESSO	55
5 SOFTWARE DE CONTROLE	57
5.1 RESET E INICIALIZAÇÃO.....	57

5.2 FLUXO DO PROGRAMA	59
5.3 INTERRUPÇÃO.....	60
5.3.1 Temporizadores.....	60
5.3.2 Input Capture	61
5.3.3 Output Compare.....	61
5.3.4 USART Portas Seriais	62
5.4 COMUNICAÇÃO	62
5.5 PROTOCOLO	63
5.6 CONTROLADORES	65
5.7 ARMAZENAMENTO DE DADOS.....	68
5.8 SISTEMA DE MONITORAMENTO.....	69
6 APLICAÇÃO AO MODELO DE ASA FIXA	72
6.1 ASPECTOS GERAIS	72
6.2 RESULTADOS DE LABORATÓRIO	72
6.2.1 Teste de Acionamento de Servo-Motores	73
6.3 RESULTADOS DE CAMPO	84
6.3.1 Coeficientes de Offset	84
6.3.2 Coeficientes Controlador PI Pitch.....	85
6.3.3 Coeficientes Controlador PI Roll.....	85
6.3.4 Teste de rota.....	86
7 CONCLUSÃO.....	92
8 BIBLIOGRAFIA	94
9 ANEXO	96
9.1 SENTENÇA NMEA183 GPGGA	96
9.2 FUNÇÃO DE INTERRUPÇÃO TIMER1	97
9.3 COORDENADOR DE DIREÇÃO.....	98
9.4 CRIAÇÃO DE SUBPONTOS.....	99
9.5 ARQUIVO DE CONFIGURAÇÃO DO SIMULADOR XML	100
9.6 PLACA VERSÃO 1.2.....	102

1 INTRODUÇÃO

Os veículos aéreos não tripulados, também conhecidos como VANTS, foram desenvolvidos inicialmente para uso militar em voos de reconhecimento e patrulha, mas estão se tornando cada vez mais presentes nas sociedades civil e acadêmica, tanto para uso em pesquisas como monitoramentos urbanos e rurais, aumentando cada vez mais seu potencial de exploração e grau de evolução a custos acessíveis à sociedade civil.

Várias aplicações ou pesquisas requerem o uso de aeronaves tripuladas, o que aumenta consideravelmente o custo de operação, implantação ou pesquisa. Com a utilização de VANTS, o custo da hora de voo pode ser reduzido consideravelmente em relação à aviação convencional, além da possibilidade de uso em voos de alto nível de periculosidade, eliminando a possibilidade de perda de vidas humanas e a necessidade de equipamentos de alto custo, tornando viáveis projetos de pesquisas e desenvolvimento de novos produtos.

Existem várias plataformas em desenvolvimento, tais como: dirigíveis, helicópteros, planadores, multirotores e aviões, aumentando cada vez mais o leque de possibilidades como monitoramento e pesquisas aquáticas, observações de cavernas e crateras vulcânicas, monitoramentos de grandes áreas agrícolas com alto grau de confiabilidade, pesquisas climáticas com maior número de amostragem em grandes áreas e em vários níveis de altitude, sistemas de busca e salvamento, controle de tráfego, supervisão de conflitos, monitoramento de grandes obras, inspeções de linhas de transmissão, observação de grandes estruturas, transporte de materiais para plataformas de produção de petróleo, mapeamento e inspeção de áreas financiáveis para cultivo.

1.1 Objetivos

As linhas de transmissão de energia elétrica exigem inspeções periódicas para avaliar, dentre outras coisas: o estado físico de estruturas, cabos e suas conexões, isoladores de suspensão e ancoragem e cabos para-raios. Inspecciona-se ainda a faixa de servidão, as vias de acesso à linha de transmissão, sempre com o objetivo de evitar interrupções do fornecimento de energia. Essas inspeções exigem o deslocamento periódico de equipe e ferramental especializados.

Uma ferramenta que pode auxiliar na manutenção preventiva e corretiva das linhas de transmissão é o uso de imagens aéreas das linhas, obtidas por meio de um VANT de asa fixa

(com câmera de vídeo) e um VANT de asa móvel (com câmera de vídeo e termográfica). As imagens podem ser usadas para a identificação de pontos com temperaturas acima do esperado e para a identificação em tempo real de possíveis problemas na linha e suas imediações e ainda para a identificação de problemas em análises posteriores de imagens armazenadas em memória. Os dois sistemas podem operar de forma autônoma, guiados por coordenadas GPS fornecidas pelo operador, ou assistida, através de um sistema de monitoramento e controle.

Dessa forma, o objetivo do projeto apresentado nesta dissertação é o desenvolvimento (projeto e implementação) de uma placa controladora, também denominado de piloto automático, para aeromodelos, e a sua aplicação em aeromodelo de asa fixa e de asa móvel (rotativa), a serem usados nas manutenções corretiva e preventiva de linhas de transmissão.

1.2 Motivação

No Brasil o sistema de abastecimento de energia elétrica é todo interligado por linhas de transmissão fornecendo energia a uma população de aproximadamente 180 milhões de habitantes. A Tabela 1 mostra a evolução da extensão das linhas de transmissão e a Figura 1, o sistema interligado (ONS, 2011).

Tabela 1: Extensão das Linhas de Transmissão do SIN – km

Tensão	2010	2011	Var % 11/10	% por tensão
230kV	43.184,5	45.708,7	5,85	44,2%
345kV	10.060,5	10.061,9	0,01	9,7%
440kV	6.670,5	6.680,7	0,15	6,5%
500kV	34.356,2	35.003,4	1,88	33,9%
600kv CC	3.224,0	3.224,0	-	3,1%
750kV	2.683,0	2.683,0	-	2,6%
SIN	100.178,7	103.361,7	3,18	100%

Fonte: ONS 2011

As linhas de transmissão do sistema interligado exigem manutenção periódica pontual, em cada torre. Dependendo da exposição às condições de temperatura, carga e tração da rede por ventos e tempestades, o período de inspeção pode variar, demandando um grande número de pessoal especializado, em condição de trabalho por vezes insalubre, somando-se ao fator

de custo elevado. Vislumbrou-se a possibilidade de desenvolvimento de uma tecnologia capaz de reduzir o trabalho em campo das equipes de inspeção e, consequentemente, reduzir o custo e a periodicidade das inspeções, aumentando assim a segurança do sistema como um todo.

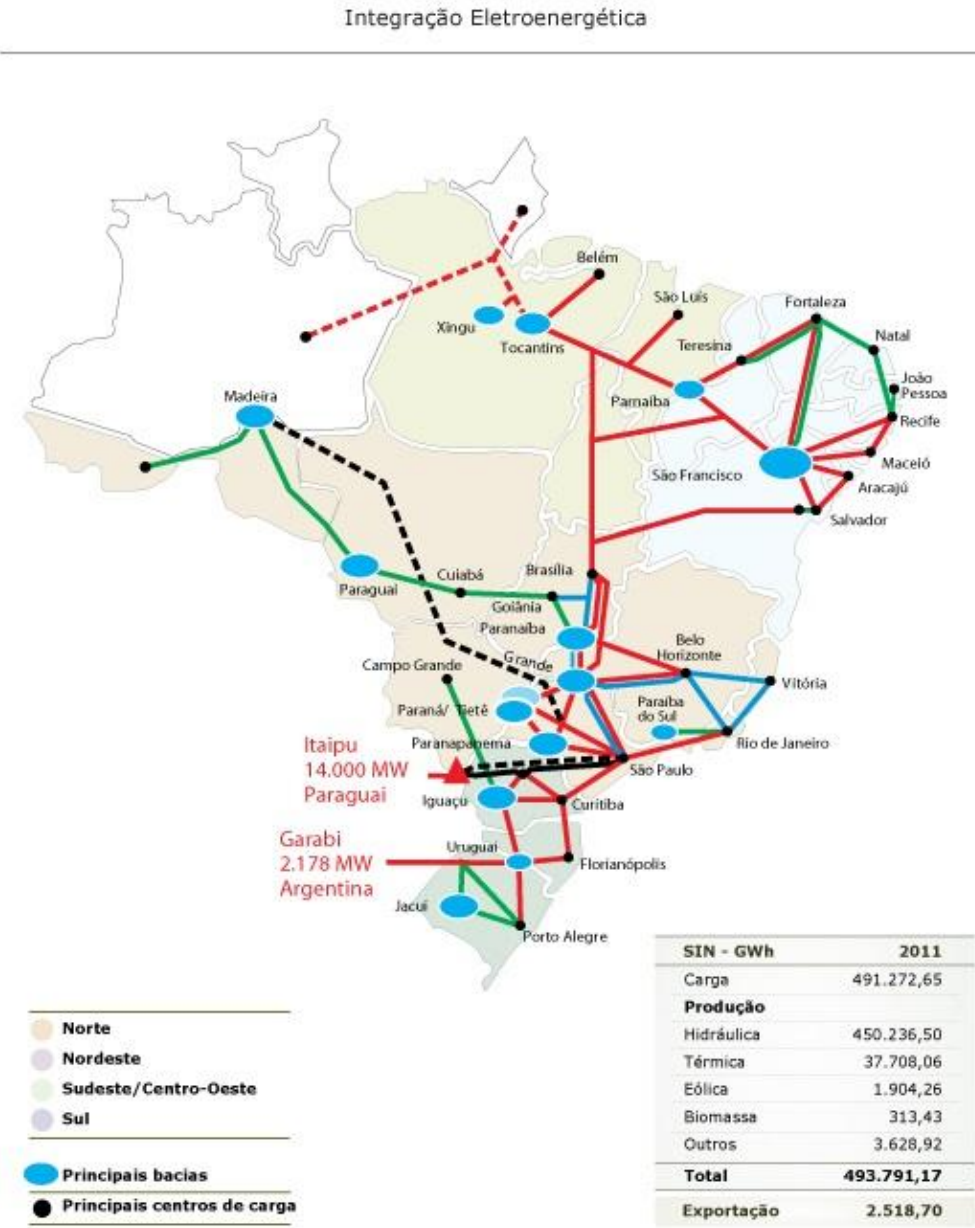


Figura 1: Mapa de interligação dos sistemas ONS 2011

Diante desse contexto, a motivação é a possibilidade de desenvolvimento de tecnologia nacional que possibilite a adequação, evolução e compatibilidade de aplicação em

que se utilize de aeromodelos embarcados do sistema de controle, comunicação e sensoriamento dedicados a operar de modo autônomo ou assistido, e executar as inspeções nas linhas de transmissão, tanto em atividades de manutenção corretiva quanto preventiva.

1.3 Organização do texto

A introdução desse trabalho apresenta os objetivos gerais do projeto e a motivação para o seu desenvolvimento.

O segundo capítulo trata da revisão bibliográfica, onde os trabalhos de pesquisa mais recentes na área são apresentados. Apresenta-se, ainda, no segundo capítulo, uma visão sobre microcontroladores, sistemas embarcados, tecnologias de construção dos sensores e suas aplicações em veículos aéreos não tripulados.

O terceiro capítulo apresenta os princípios aerodinâmicos dos aeromodelos de asa fixa, que servem como base para o sistema de controle dos aeromodelos. No quarto capítulo apresenta-se o desenvolvimento da placa controladora, destacando o funcionamento e a importância de cada um dos sensores utilizados.

O quarto capítulo descreve o projeto e a construção de uma placa com uma versão inicial para testes e aperfeiçoamento, e uma versão final com todas as correções necessárias ao funcionamento final para o propósito específico de inspeção linhas de transmissão.

O quinto capítulo descreve o software da placa controladora e todos os subsistemas necessários para o correto funcionamento do sistema, e o sistema de monitoramento.

O Sexto capítulo apresenta os detalhes da implementação da placa no modelo de asa fixa e seus resultados, de laboratório e de campo.

Na conclusão são feitas considerações sobre o desenvolvimento do trabalho e seus resultados e ainda são apresentadas propostas para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

2.1 Histórico

Os primeiros veículos aéreos não tripulados (VANTS) que obtiveram êxito são do início do século XX e eram torpedos aéreos usados na primeira guerra mundial. Na época já houve a tentativa de construção de controles eletrônicos, porém a capacidade tecnológica limitada inviabilizou seu uso devido ao peso e à baixa precisão. Entre as décadas de 40 e 50 adaptou-se um avião em um VANT rádio controlado (OLIVEIRA, 2010), porém a baixa qualidade de imagem dificultava a identificação do alvo.

Ao longo de décadas o incentivo ao desenvolvimento de VANTS tem sido uma busca fundamentalmente para fins militares, a exemplo de mísseis teleguiados. Ao redor do mundo tem-se observado uma corrida para o desenvolvimento de tecnologia e construção de VANTS para aplicações civis. Os EUA é o país com maior nível de investimentos e domínio desta tecnologia. O Japão se destaca também na produção e desenvolvimento de VANTS, em especial como o país em que há um grande emprego de VANTS na agricultura, inclusive com equipamentos para pulverização (JORGE, ANDRE, & INAMASU, 2011). Outros países também têm projetos próprios para aplicações civis e militares, inclusive com disponibilidade de exportação.

Avanços recentes nas áreas de tecnologia e capacidade computacional, de criação de compostos mais leves e resistentes, além da melhoria dos sistemas de posicionamento baseados em sistemas de posicionamento global (GPS), a industrialização de micro mecanismos em escalas de nanômetros, têm impulsionado a expansão deste seguimento para uso civil (JORGE, ANDRE, & INAMASU, 2011). No Brasil os primeiros relatos sobre a construção de VANTS juntamente com o desenvolvimento do sistema de controle datam da década de 80 com o projeto ACAUÃ, pelo Centro de Pesquisas Aeroespaciais (CTA), voltado para uso militar; projeto este que ficou parado do ano de 1988 até o ano de 2007 (ESCOSTEGUY, 2009). A Empresa Brasileira de Pesquisas Agrárias (Embrapa) patenteou em 2004 o projeto ARARA, mas nunca chegou a completar todos os objetivos traçados; e após várias quedas a Embrapa parou de participar do projeto e a versão patentada foi utilizada na AGX tecnologia para fins comerciais (JORGE, ANDRE, & INAMASU, 2011).

Em 2009 a Cemig, juntamente com a Fitec, deu início ao projeto de um VANT destinado ao monitoramento de linhas de transmissão, demonstrando as grandes

possibilidades de aplicação destes equipamentos (AMARAL, 2013). Neste mesmo nicho de aplicação a CELG - Distribuição, juntamente com a Universidade Federal de Goiás (UFG) iniciou um projeto nos mesmos moldes de aplicação, mas com foco na construção de uma placa de controle compatível com o ambiente de atuação, para monitoramento de linhas de transmissão, prevendo o uso de estruturas de asa fixa e de asa móvel.

2.2 Microcontroladores e Aplicações em VANTS/Sistema Embarcados

Um microcontrolador é a soma de componentes distintos para constituir um sistema de processamento computacional em um único chip. Os mais simples são dotados apenas de Unidade Central de Processamento (CPU), memória de programa, memória de dados, barramentos de comunicação interna, sistema de clock e portas de entrada e saída.

Os microcontroladores podem ser de uso específico, sendo dotados de características funcionais específicas, ou mesmo acrescidos com periféricos para uma maior abrangência de aplicações como conversores Analógicos Digitais (A/D), temporizadores, canais de comunicação serial e/ou sistemas de interrupção.

Os sistemas de controle para sistemas embarcados, como um todo, há muito tempo utilizam cada vez mais técnicas avançadas de filtragem de sinais, para amostragens em períodos cada vez menores, visando uma resposta mais precisa. Para tal o sistema computacional é sempre requerido quase em sua totalidade pela demanda de processamento e integração com interfaces analógicas. A integração de dados analógicos, somado ao processamento em tempo real requereu da indústria um novo conceito de componente para processamento e integração do sistema como um todo: o Processador Digital de Sinais (*Digital Signal Processor* - DSP).

2.3 Processadores Digitais de Sinais (DSP)

Os processadores digitais de sinais são muito semelhantes aos microcontroladores em sua arquitetura de funcionamento, tendo os mesmos elementos básicos como CPU, memória de dados, memória de programa e periféricos, inclusos em um único chip (Introdução ao DSP, 2002) e são acrescidos de funcionalidades para propósitos diversos aumentando o desempenho.

Esse desempenho superior é devido a:

- Multiplicação entre registro e o acumulador em um único ciclo. Alguns DSPs de alta performance possuem mais de um multiplicador, que permite realizar duas ou mais operações ao mesmo tempo por ciclo de instrução, sendo ainda que alguns possuem hardware para operações matemáticas em ponto flutuante.

- Modos específicos de endereçamento: com pré e pós-modificação dos ponteiros de endereços, endereçamento circular e endereçamento do tipo bit-reversed.

- Arquitetura de múltiplo acesso à memória, que possibilita a leitura e escrita na memória on-chip em um único ciclo.

- Instruções especiais de controle de execução, como por exemplo, instrução de loop, não necessitando assim instruções de atualização e teste de contadores de loop.

- Conjunto irregular de instruções, o que geralmente possibilita mais de uma operação em um único ciclo de instrução (como exemplo, em um DSP com arquitetura de 32 bits, 16 bits podem ser destinados a operações diversas e o restante para ponteiro dos dados resultantes).

Com a evolução nos processos de fabricação de chips, os DSPs vêm ganhando incrementos significativos no poder de processamento, tanto pelo incremento em velocidade como na expansão das arquiteturas de processamento. Observa-se a oferta de DSPs de 32bits em chips de 4x4x1,5 mm direcionados ao tratamento de processos antes predominantemente desenvolvidos com o uso de microcontroladores de 8 bits.

Os DSPs são mais indicados onde há a necessidade de processamento em tempo real (os dados são tratados e processados antes de uma nova amostragem) e onde há uma maior complexidade nas operações aritméticas.

2.3.1 Aplicações de Microcontroladores e DSPs

Com a otimização dos processos de fabricação reduziu-se consideravelmente os custos de produção de DSPs e microcontroladores, proporcionando seu uso em produtos de valor final reduzido, considerando que o custo de um chip dedicado pode se tornar mais caro, se for considerado todo o processo de fabricação e desenvolvimento dedicado a uma função ou produto. Assim, hoje praticamente todo equipamento que dependa de alguma lógica computacional utiliza DSP ou microcontroladores.

Na indústria, está presente nos mais diversos tipos de CLPs (Controlador Lógico Programável) e sistemas supervisórios, eliminando em muitos equipamentos a necessidade de

computadores pessoais devido ao elevado nível de processamento do DSP. Em residências, nos automóveis, no cotidiano como um todo, os microcontroladores e DSPs são usados e em grandes escalas em equipamentos como controle remoto de TVs, a própria TV, o controle de portão eletrônico, sistemas de segurança, controles de acesso, gravação de imagens em tempo real, e toda uma vasta gama de produtos em que a necessidade de processamento demandava o uso de um computador pessoal, nos dias de hoje podem se beneficiar da evolução dos DSPs.

Em sistemas embarcados para VANTs o sistema de processamento requer interfaceamento com diversos sensores para a integração.

2.3.2 DSPIC Características Gerais

O DSPIC33EP512MC806, DSP produzido pela Microchip Technology Inc., é um componente integrado que contém, em um único dispositivo, todos os circuitos necessários para processamento e várias interfaces periféricas. Esse dispositivo é mostrado na Figura 2: DSP Microchip 16bits, e é utilizado no projeto vant CELG Distribuição.

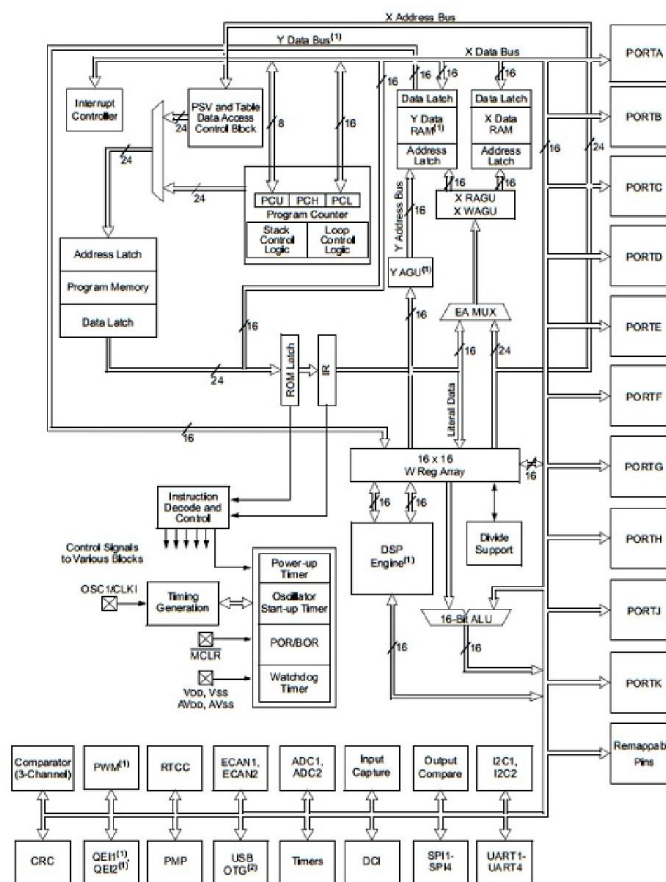


Figura 2: DSP Microchip 16bits (MICROCHIP T. , 2011)

Uma característica da família de DSP da Microchip é a alta integração de interfaces periféricas, que resulta em uma redução no tamanho final do produto por dispensar a necessidade de interfaces externas nas quais pode-se citar algumas utilizadas na placa controladora.

2.3.2.1 Reset

O DSPIC33EP512M possui internamente um controle de reset conforme a Figura 3 em que se destaca alguns recursos como controle de tensão mínima para funcionamento, power-up, execução de instrução com endereçamento errado, relógio decrescente de reset (Watchdog), pino de reset, conflito de interrupção, dentre outros recursos que podem ser utilizados. Apesar de não ser comumente usual, a correta utilização destes recursos aumenta consideravelmente a estabilidade do software da aplicação. No projeto apenas o watchdog não é habilitado na fase inicial no projeto Vant.

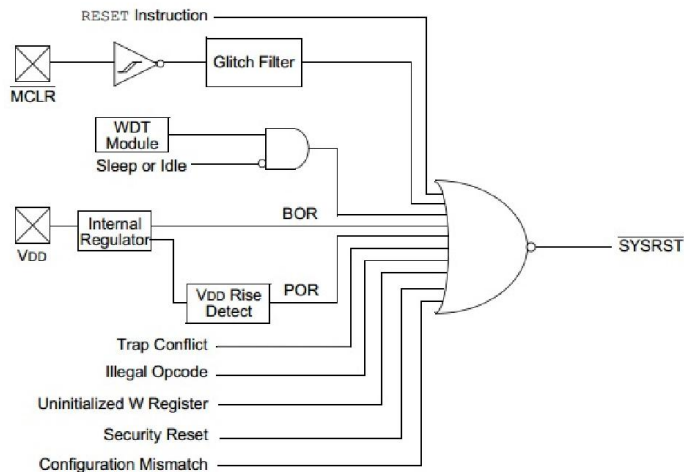


Figura 3: Controlador de reset (MICROCHIP T. , 2011).

2.3.2.2 (UART)– Universal Asynchronous Receiver Transmitter.

Interface serial full duplex de 8 bits ou 9 bits de dados para endereçamento, beneficiando protocolos de controle mestre/escravo. A interface está preparada também para controle por hardware com pinos dedicados: UxCTS e UxRTS conforme a Figura 4, além da

interconecção IrDA com o codificador e o decodificado, registrador First-In First-Out (FIFO) de 4 bytes para transmissão e para recepção Figura 4.

A interface serial possui interrupções tanto para transmissão quanto recepção, que podem ser habilitadas de forma independente para eventos de 1 a 4 bytes, de acordo com o preenchimento da FIFO. Com a utilização da FIFO pode-se reduzir em até 4x o número de interrupções.

O Baud Rate pode variar entre 17.5 Mbps a 67 Mbps para o clock de 70 MIPS, e a interface serial pode ainda ser configurada em modo Loopback. Neste modo de operação os dados de saída são encaminhados diretamente para entrada e, deste modo, é possível confirmar o bom funcionamento. No projeto da placa estas interfaces são responsáveis pelo recebimento e envio de dados ao rádio comunicador, GPS e GPRS.

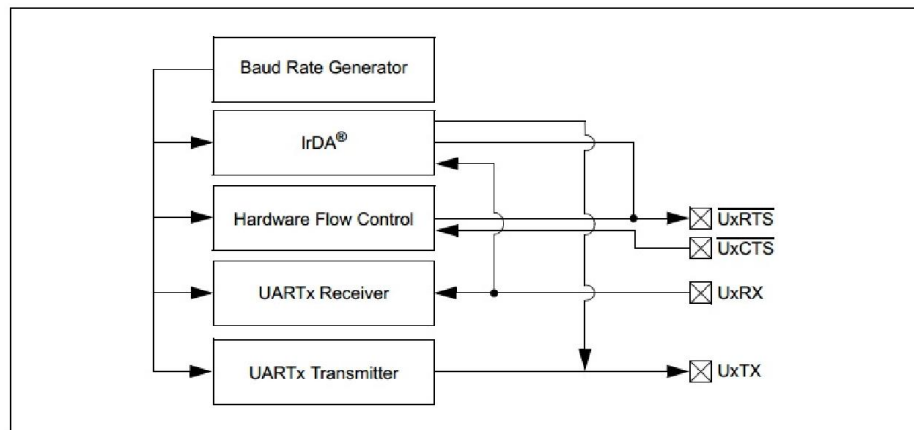


Figura 4: Pinos de conexão da porta serial (MICROCHIP T. , 2011)

2.3.2.3 Temporizador - (TIMER)

O DSPIC33EP512MC806 possui 9 temporizadores de 16 bits, sendo o timer1 um temporizador diferenciado, pois possui configuração para oscilador externo de baixo consumo podendo assim usar o temporizador como Contador Real de Tempo (RTC), conforme mostrado na Figura 5.

Os temporizadores de 2 a 9 são contadores de 16 bits que podem ser concatenados em pares para uma configuração de 32 bits. Os temporizadores podem ainda ser configurados como contadores ou temporizadores síncronos para contagem de eventos externos. Os temporizadores servem de referência para outras interfaces como *Input Capure (IC)*, *Output*

Compare (OC), podendo ainda ser usados como disparadores síncronos para outras interfaces, como ADC, e Interrupções.

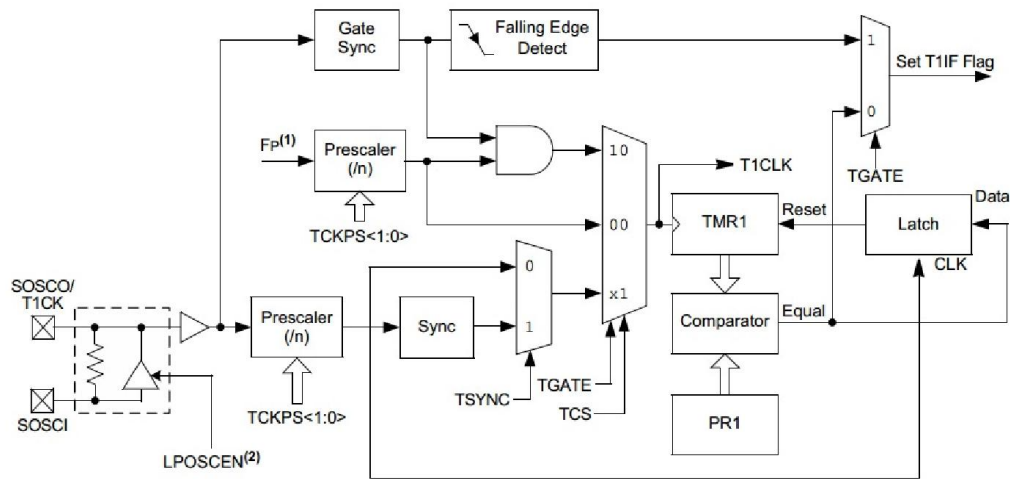


Figura 5: Temporizador 1 com oscilador de entrada (MICROCHIP T. , 2011).

Os temporizadores podem ainda ser configurados como contadores de eventos externos com as mesmas funcionalidades de clock interno. No projeto da placa são utilizados 5 temporizadores, todos em modo temporizador de 16 bits sendo um para disparo de interrupção a cada 1 ms, o restante para referência de tempo para outras interfaces como IC, OC e ADC.

2.3.2.4 Serial Peripheral Interface (SPI)

O DSPIC33EP512MC806 possui 4 interfaces Serial Peripheral Interface (SPI). Essas interfaces são síncronas com alta velocidade de transferência, podendo chegar a 15 Mbps. Neste chip a interface pode se comunicar em 8 ou 16 bits de dados, e pode necessitar de 2, 3 ou 4 pinos, de acordo com o modo de operação. Há também FIFO de 8 níveis tanto para entrada quanto para saída, conforme Figura 6.

A interface SPI pode funcionar como mestre ou escravo de acordo com a configuração aplicada. Possui vários modos de interrupção, desde o preenchimento total da FIFO à recepção ou transmissão de um byte.

O uso da interface SPI neste projeto é voltado ao interfaceamento do cartão MMC, utilizando software de controle para emular uma interface entre cartão SD e DSP, ampliando

assim a capacidade de armazenamento. No projeto, esta interface é inicializada em 8 bits com taxa de transferência de 400 kbps, e após a inicialização do cartão, a taxa sobe para 10 Mbps.

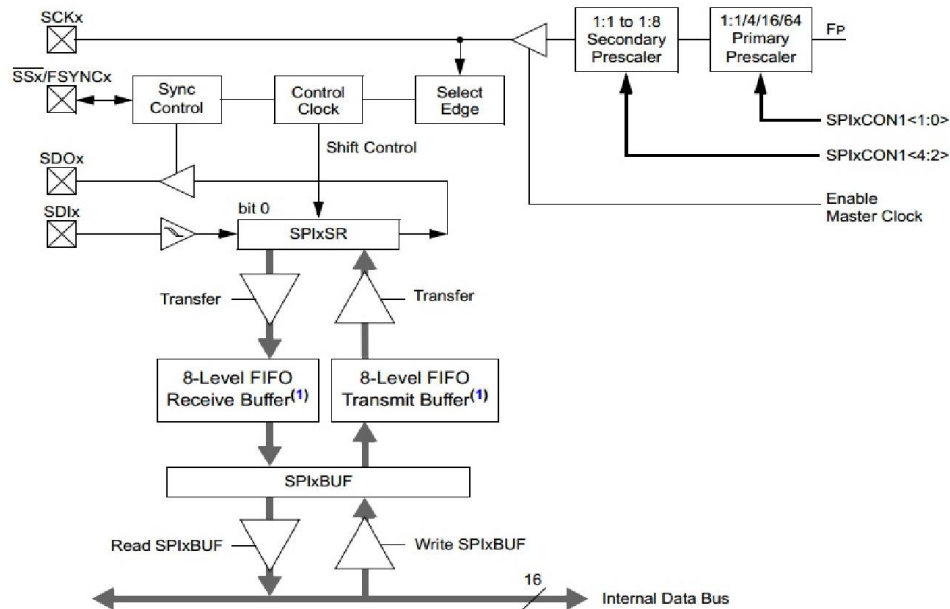


Figura 6: Interface SPI (MICROCHIP T. , 2011).

2.3.2.5 Inter-Integrated Circuit

A interface Inter-Integrated Circuit, IIC, ou I2C, diferente das outras interfaces de comunicação em que se interligam 2 componentes diretamente, a I2C é na verdade um barramento em que se pode conectar vários componentes em paralelo, desde que os mesmos respondam através de endereços distintos, possibilitando assim o interfaceamento de diversos equipamentos por apenas 2 pinos considerando um componente mestre e outros escravos, podendo haver mais de um componente mestre.

Na comunicação I2C é informado o endereço do dispositivo, a direção dos dados, o endereço do registro e, posterior a isso, os dados. Para muitos tipos de dispositivos, tais como memórias, o endereço de registro trabalha em páginas onde se informa a página, sendo necessário informar a página (High Byte) e o registro (Low Byte), conforme Figura 7.

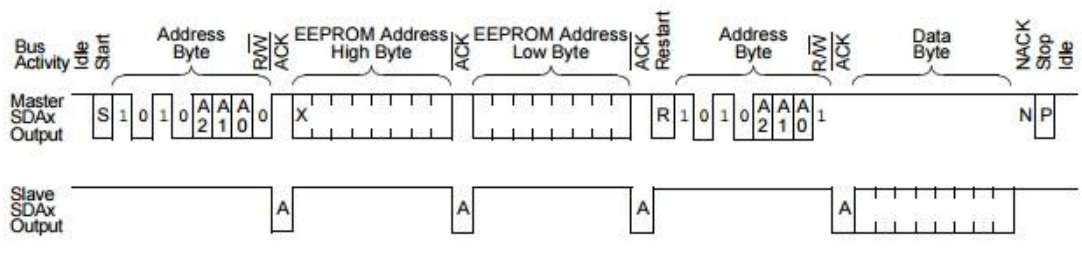


Figura 7: Comunicação I2C com memória (MICROCHIP T. , 2011).

O DSPIC33EP512MC806 possui 2 interfaces I2C que podem ser configuradas como mestre ou escravo, tem endereçamento de 7 ou 10 bits, velocidade de 1 Mbps, possui vários modos de interrupção, os pinos de interconexão das interfaces são dedicados e não podem ser alterados por endereçamento interno. A estrutura interna da interface pode ser observada na Figura 8, onde é possível observar que cada bloco interno da interface é conectado em paralelo com o barramento, para que o barramento possa ser conectado diretamente a vários circuitos sem que haja bloqueio de sinais por interligar vários pinos de saída a um mesmo ponto, adotou-se por padrão de interconexão saídas em coletor aberto, desta forma é possível errar a comunicação ao tentar acessar o barramento por mais de um mestre, mas o circuito não seria danificado.

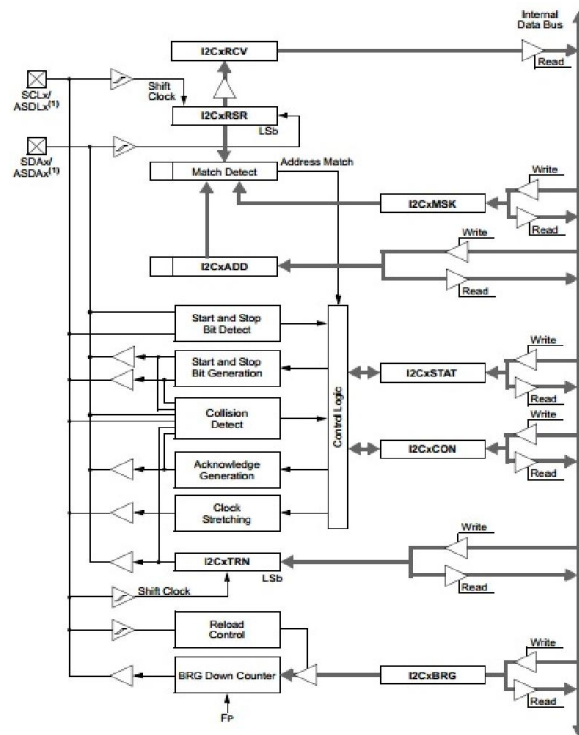


Figura 8: Interface I2C (MICROCHIP T. , 2011).

Os pinos de cada dispositivo conectado ao barramento I2C trabalha com baixa impedância apenas para o nível lógico 0, assim, por mais que qualquer componente ligado ao barramento trave, ou até mesmo esteja danificado, o barramento apenas cessa a comunicação sem maiores danos. No projeto da placa os sensores inerciais, o sensor magnetômetro e a memória não volátil são conectados a este barramento.

2.3.2.6 Conversor Analógico-Digital (ADC)

Interface ADC de 10 bits com 1.1 Msps ou 12 bits com 500 Ksps, referência de entrada, entradas diferenciais, 4 canais independentes, 32 portas de entrada, sample hold para amostragem, buffer de 256 words (16bits) Figura 9.

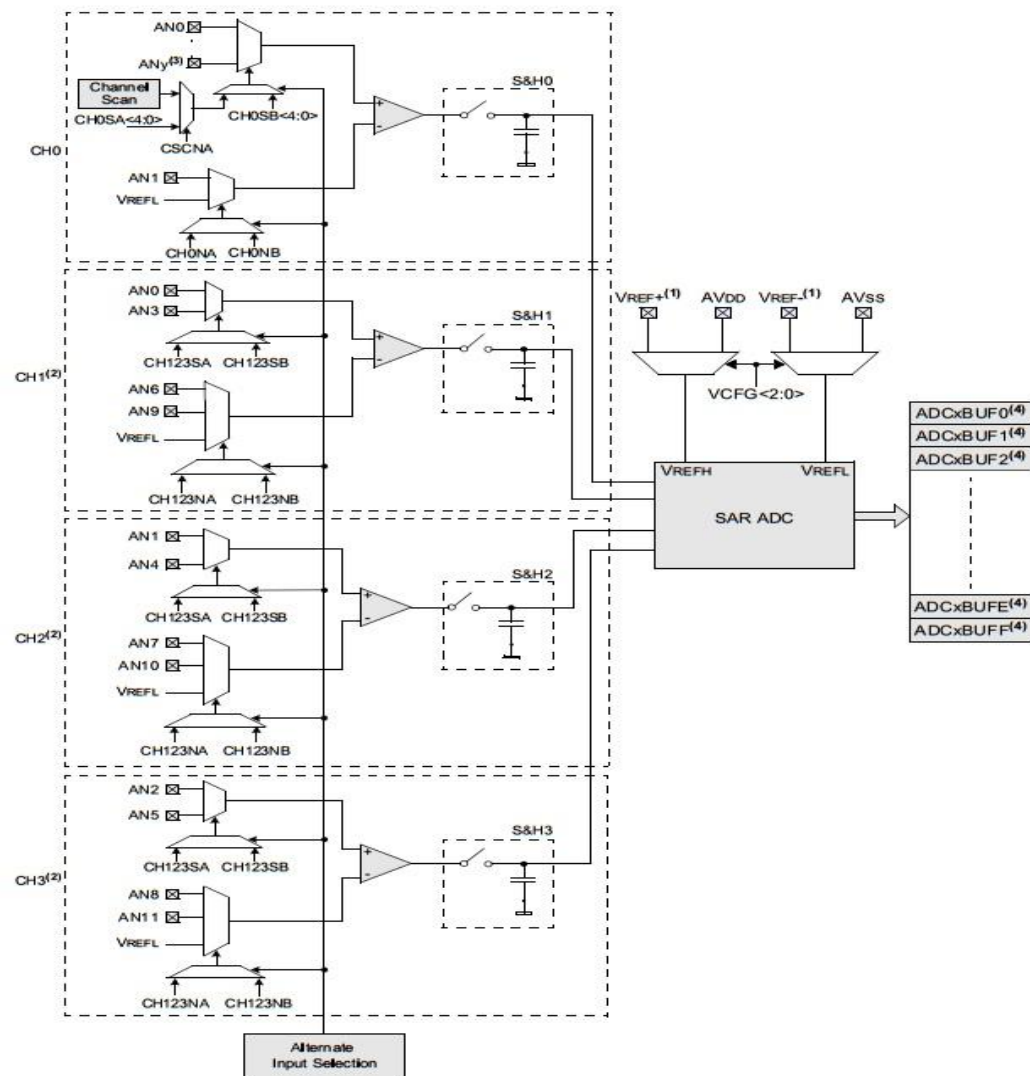


Figura 9 Interface ADC (MICROCHIP T. , 2011).

A interface possui varredura automática dos canais selecionados, modos de disparo associado a outras interfaces. No projeto da placa o ADC é utilizado na leitura dos valores de tensão e corrente de bateria, sensor de velocidade (tubo de pitot) e são reservadas entradas para possibilidades futuras.

2.3.2.7 Input Capture (IC)

Muitos dos sinais de entrada utilizados em controle são analógicos, sendo condicionados pela largura de pulso ou posição do pulso, tais como sinais de modulação PWM e/ou PPM, para ler tais dados devemos medir com precisão a largura destes pulsos no DSPIC e utilizado uma interface de captura de pulsos, a interface pode ser observada na Figura 10.

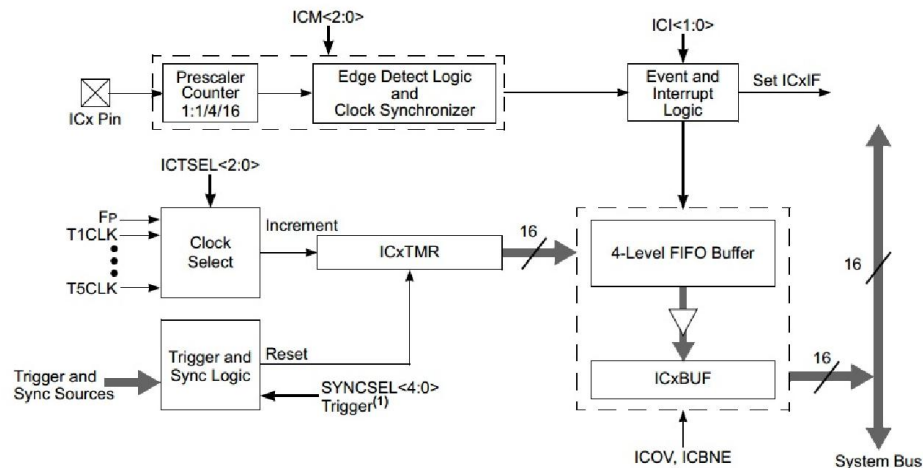


Figura 10: Input Capture (MICROCHIP T. , 2011).

A interface *Input Capture* (IC) do DSPIC33ep512 utiliza como referência um dos *timers* para medir os eventos na entrada, verificando a largura ou período de qualquer tipo de sinal pulsado, sendo de grande importância para leitura de vários sinais simultâneos sem demandar processamento da CPU do DSP. A interface possui vários modos de interrupção para disparo de contagem e um buffer do tipo FIFO de 4 níveis. Sua aplicação reduz consideravelmente o uso de interrupções, uma vez que a mesma é responsável por capturar os valores das saídas do receptor de controle remoto, e decodificar os dados serializados do sinal PPM.

2.3.2.8 PPS Peripheral Pin Select (PPS)

Um recurso presente no DSPIC33EP512 e pouco apresentado por outros componentes é o PPS, do inglês, *Peripheral Pin Select*, recurso este que permite alterar a posição dos pinos de saída ou entrada de praticamente todas as interfaces, excluindo apenas interfaces de entrada analógica como ADC, comparadores de tensão ou interfaces de barramento.

Este recurso permite que a placa de circuito seja desenhada com um visual mais limpo, com menos ilhas de passagem para outro layout. No projeto da placa este recurso foi amplamente utilizado para reduzir o número de vias entre as camadas da placa.

Após o reset do DSP o programa interno desbloqueia uma chave, altera os pinos e bloqueia novamente. Esta chave pode ser configurada para ser desbloqueada uma única vez após reset, ou sempre que necessário. Esta definição de modo de destravamento é feita em tempo de compilação nos bits de configuração do DSP.

2.3.2.9 Output Capture (OC)

Alguns sistemas de controle ou comunicação trabalham com sinais pulsados, tais como controladores de motor ou controladores de temperatura, que na sua grande maioria correspondem a sinais com largura de pulso variável, *Pulse Width Modulation* (PWM). A geração de sinais em frequências elevadas pode comprometer grande parte do processamento de uma CPU por ser necessário associar a um sistema de interrupção por timer.

No DSPIC está presente a interface *Output Capture* (OC), que gera estes sinais sem a intervenção da CPU e utilizando como base apenas a contagem de um dos temporizadores. Essa interface é mostrada na Figura 11.

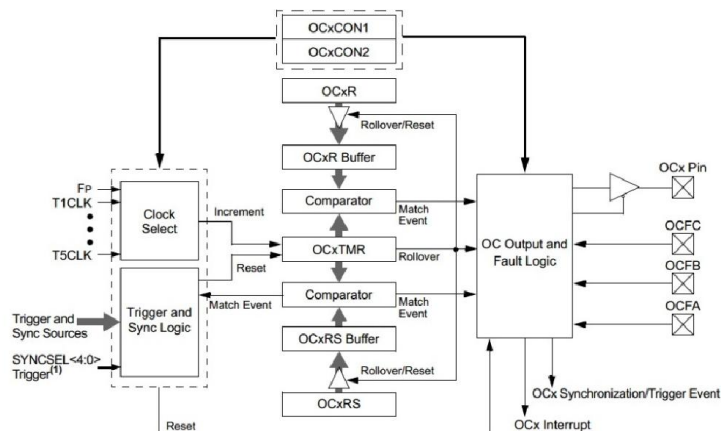


Figura 11: Output Capture (MICROCHIP T. , 2011).

De forma similar à interface IC esta interface é de grande relevância no projeto da placa por ser responsável por gerar o sinal de controle para os servos.

2.4 Sistemas Micro-Eletromecânicos (MEMS)

Sistemas Microeletromecânicos ou MEMS (*Micro-Electro-Mechanical System*) foi o termo criado no fim dos anos 80 nos Estados Unidos para descrever sistemas miniaturizados compostos basicamente de um sensor, que insere informações no sistema, um circuito eletrônico que trata o sinal do sensor e um atuador que responde ao sinal gerado pelo circuito eletrônico (CALACHE, 2013).

Os sensores baseados em MEMS vêm substituindo alguns dos seus precursores, pois apresentam menor tamanho, menor peso, menor consumo de energia, menor custo e alta confiabilidade de operação, quando comparados com os sensores convencionais, e a principal característica que ainda favorece os sensores convencionais é a precisão das medições que ainda supera a precisão obtida com os dispositivos MEMS, mas esta diferença vem se tornando cada vez menor (SOARES DE LIMA, 2005).

Grandes investimentos para o desenvolvimento de MEMS ou da Tecnologia de Micro Sistema (*Micro System Technology* – MST terminologia europeia) vêm tornando possível a integração em larga escala de sensores em espaços cada vez menores por meio da utilização generalizada dos recursos e da experiência bem sucedida na produção de circuitos eletrônicos integrados, embora a ideia de construir dispositivos microscópicos não seja recente, somente com os desenvolvimentos da microeletrônica foi possível iniciar a produção destes sistemas (SOARES DE LIMA, 2005).

Enquanto a microeletrônica é composta por estruturas sólidas e compactas, MEMS possuem cavidades, canais, membranas e outros elementos incomuns a dispositivos eletrônicos. Na Figura 12 pode-se observar um conjunto de engrenagens embaixo de um ácaro. A partir da década de 1980, diversas tecnologias de deposição e micro usinagem foram desenvolvidas ou aprimoradas com o objetivo de viabilizar a produção em escala industrial destes dispositivos. Os processos básicos são a deposição de camadas de substratos diferentes, modelagem por fotolitografia e usinagem (WEBER, 2014).

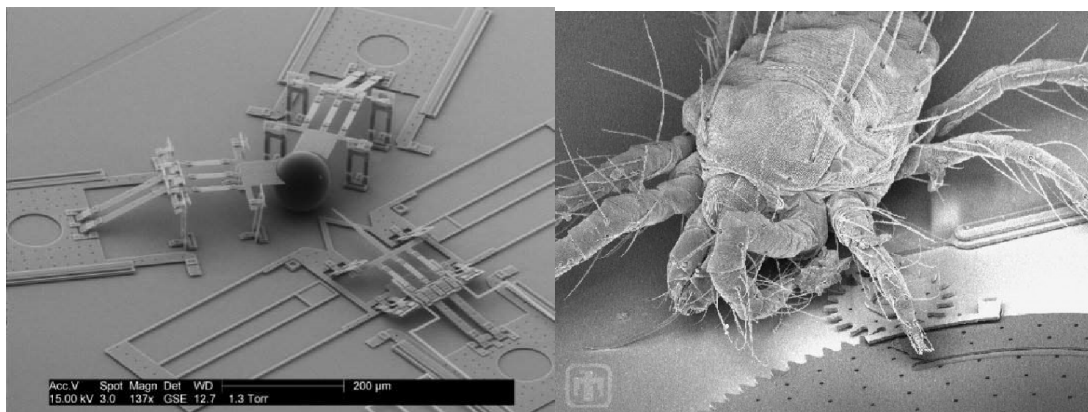


Figura 12: Nano injetor (fonte: Brian Jensen/BYU) e um ácaro sobre estrutura (fonte: SOARES DE LIMA, 2005)

Embora materiais como polímeros, cerâmica e metais também sejam utilizados, o substrato de silício é o mais utilizado, como resultado da ampla experiência com o material adquirida com a microeletrônica. Os dispositivos MEMS dividem-se basicamente em seis categorias: sensores de pressão, sensores inerciais (acelerômetros, giroscópios, magnetômetros e outros), microdispositivos para fluidos, análises clínicas, óptica e RF, cujos principais fabricantes são Motorola, Analog Devices, Freescale, STMicroelectronics, Texas Instruments, Sensoror e Delphi (WEBER, 2014).

A tecnologia MEMs não se restringe a criação de sensores, motores, estruturas e engrenagens, dentre muitas aplicações como sensores, é utilizada na área de ciências biológicas como um nano injetor que pode ser visto na Figura 12. No projeto da placa os sensores inerciais utilizam desta tecnologia.

3 PRINCÍPIOS AERODINÂMICOS DE UM AEROMODELO.

O controle de uma estrutura como um aeromodelo de asa fixa ou asa móvel, de forma a manter essa estrutura voando e em equilíbrio dinâmico, exige o conhecimento das forças que atuam na estrutura e suas relações com grandezas tais como velocidade, aceleração, massa e os eixos de referência. Exige também o conhecimento da atuação das estruturas de controle (ailerons, flaps, leme, profundor para o sistema de asa fixa) e de vários sensores (GPS, acelerômetro, giroscópio, altímetro, magnetômetro, barômetro) e modos de interfaceamento para o controle da placa.

Assim, o presente capítulo apresenta as forças principais envolvidas na sustentação de um aeromodelo, a atuação das superfícies de controle e a metodologia de controle, a partir dos sinais dos sensores.

3.1 Sistemas de Referência para um Corpo no Espaço

Para que se possa indicar um sentido ou direção de um corpo deve-se modelar matematicamente esse corpo em relação a um sistema de referência, de maneira que a informação oriente o referido corpo no espaço. Conforme descrito, a modelagem de um sistema de navegação inercial é realizada com base na definição de diversos sistemas de referência.

Na Figura 13 estão representados graficamente quatro sistemas de coordenadas de referência dextrogiros, usuais para a utilização de observações de plataformas inerciais, cada um contendo três eixos coordenados ortogonais entre si conforme descrito em (SOARES DE LIMA, 2005).

1) Referencial Considerado Inercial ou Quase Inercial (i): eixos (X_i , Y_i , Z_i) com a origem no centro de massa da Terra e onde X_i aponta na direção do ponto vernal, Z_i coincide com o eixo de rotação médio da Terra, e Y_i completa o sistema dextrogiro.

2) Referencial Terrestre (e): eixos (X_e , Y_e , Z_e), onde a origem está no centro de massa da Terra, o eixo X_e corta o meridiano médio de Greenwich e o eixo Z_e coincide com o eixo Z_i definido anteriormente.

3) Referencial de Navegação (Nível Local) (n): onde os eixos podem ser indicados como (N, E, D) e a origem se localiza numa posição local do sistema de navegação, sendo que

o eixo N orienta-se na direção norte, o eixo E orienta-se na direção leste e o eixo D orienta-se na direção normal ao elipsoide de referência.

4) Referencial dos eixos dos sensores (do corpo ou da plataforma) (b): eixos (X_b , Y_b , Z_b), onde a origem localiza-se no centro do conjunto de sensores. Considerando-se a estrutura do veículo que transporta os sensores (a plataforma), geralmente X_b orienta-se para a frente, Y_b orienta-se para a direita e Z_b orienta-se para baixo.

No sistema da placa em desenvolvimento adota-se o referencial dos eixos dos sensores em relação ao centro de massa da terra para determinar as inclinações da placa, e referencial de navegação para o posicionamento da placa no globo terrestre.

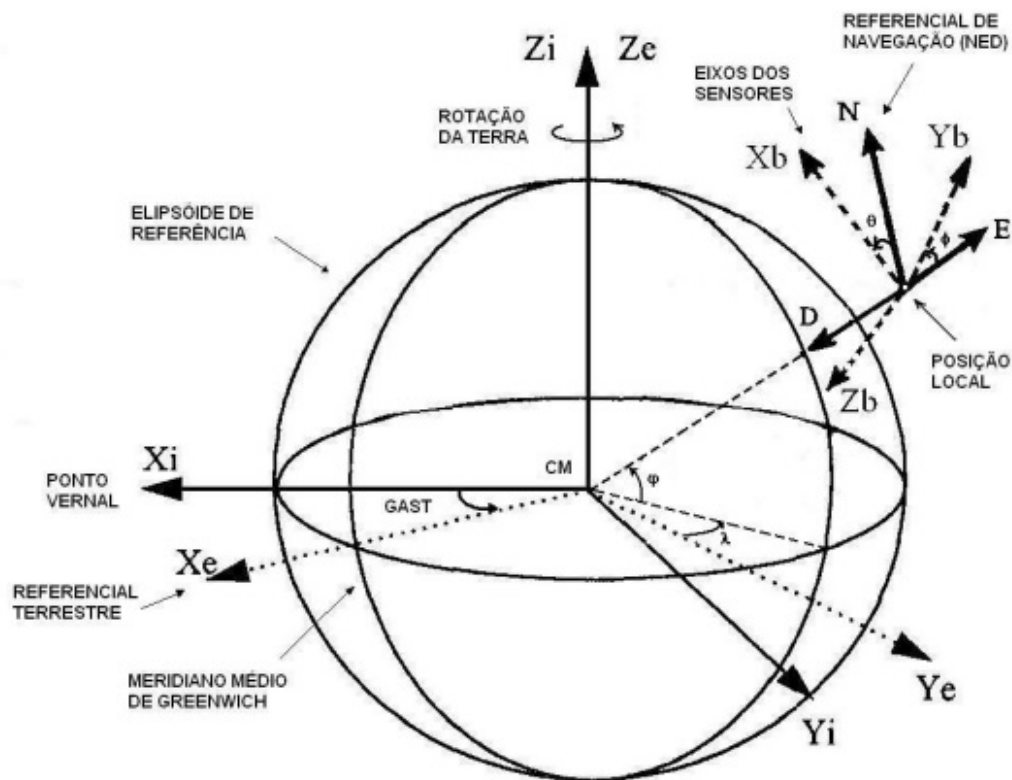


Figura 13: Eixos de referência (Fonte (SOARES DE LIMA, 2005))

3.2 Dinâmica para Modelo de Asa Fixa

De acordo com (ARAÚJO., 2007), o teorema de Bernoulli enuncia que quanto maior o fluxo de um fluido que passa por uma superfície, menor a pressão exercida sobre ela.

Assim, podemos concluir que o avião se mantém estável no ar não por ter asas, mas devido à diferença de velocidade do fluxo de ar transversal que passa pelas duas partes da asa (intradorso parte inferior da asa e extradorso parte superior da asa) gerando assim uma sustentação proporcional ao quadrado da velocidade deste fluxo conforme mostrado na Figura 14.

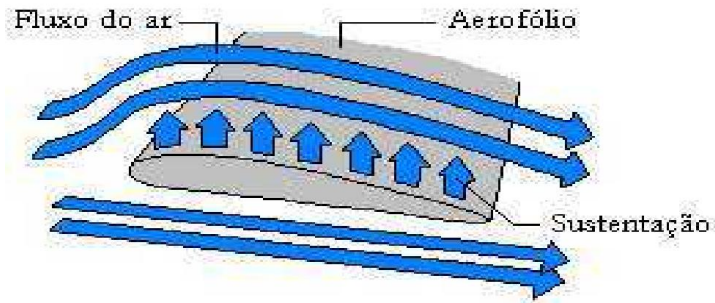


Figura 14: Fluxo de Ar Sobre a Asa (Fonte: Portal ime.unicamp)

É importante observar que o desenho geométrico da asa, chamado de perfil, varia com cada aplicação a que o avião é exposto onde um avião agrícola tem perfil diferente de um jato, sendo fundamental obter a sustentação necessária com a velocidade compatível a determinada circunstância.

Para que haja um fluxo de ar constante nas asas deve-se promover um empuxo no avião com velocidade suficiente para gerar a sustentação necessária. Este empuxo é gerado pelo conjunto motopropulsor, podendo ser um motor elétrico ou a pistão acoplado a uma hélice ou uma turbina a jato.

O empuxo gerado pela hélice segue os mesmos princípios da asa para gerar uma sustentação em relação ao intradorço, com a diferença que para compensar a velocidade angular entre a ponta da hélice e a raiz há uma maior inclinação entre o bordo de fuga e o bordo de ataque na raiz da hélice, onde a raiz da hélice tem a maior inclinação, que vai variando proporcionalmente com a velocidade angular de cada ponto até a ponta da hélice.

Quando uma superfície é submetida a um fluxo de ar observa-se uma resistência a este fluxo, no caso do avião denomina-se de arrasto o atrito gerado pelas superfícies em contato com o fluxo constante de ar.

Para que um avião se mantenha no ar em uma trajetória paralela ao horizonte pode-se então afirmar que o empuxo deve ser igual ao arrasto e a sustentação deve ser igual ao peso, conforme apresenta a Figura 15.

Conforme o (FRANCISCO, 2012), para se calcular a sustentação de uma asa pode-se usar a equação 3-1.

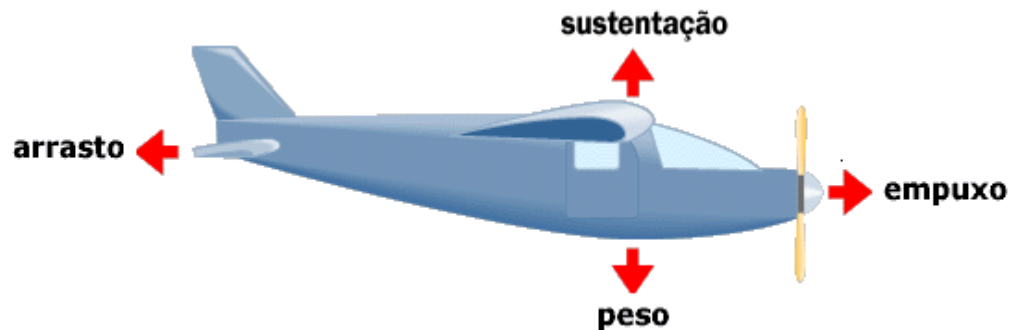


Figura 15: Forças de Atuação em Um Avião (Fonte: (FRANCISCO, 2012))

$$L = C_l \times \frac{1}{2} \times \rho \times V^2 \times A \quad [3-1]$$

Em que

L=Sustentação (N)

C_l=Coeficiente de sustentação do perfil

ρ = densidade do ar (kg/m³)

V=Velocidade (m/s)

A=Área da asa (m²)

3.2.1 Estruturas e Superfícies de Controle de um avião

Em um contexto geral um avião é composto por quatro partes básicas: fuselagem, asas, motores e superfícies de controle. Estas superfícies variam em número de acordo com o tipo de avião, porém 4 destas são praticamente universais: ailerons, profundores, leme, flaps e são localizadas, de maneira geral, em pontos idênticos. No avião elas são distribuídas conforme a Figura 16.

Asas: superfícies longitudinais que se prendem à fuselagem ou corpo do avião e que geram a maior parte da sustentação do avião. As asas podem ter vários formatos como plana, delta, com ou sem diedro. Também podem servir de base de apoio para os motores.

Fuselagem: estrutura central onde se aloja a cabine de comando e a carga útil.

Motores: partes responsáveis por gerar o empuxo necessário, geralmente os motores são instalados nas asas ou na parte dianteira do corpo do avião.

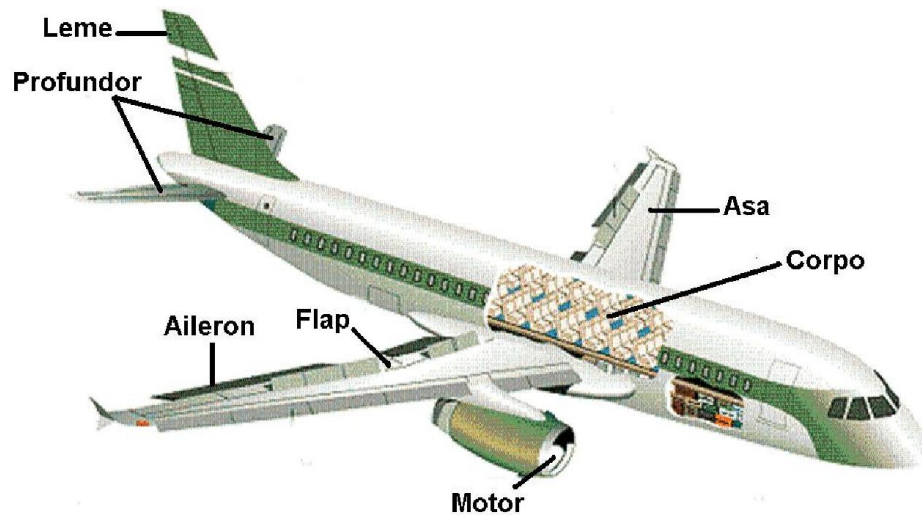


Figura 16: Estruturas Básicas de Controle um Avião (Fonte: (FRANCISCO, 2012))

- Superfícies de controle

As superfícies de controle estão distribuídas nas asas e grupo de cauda, sendo as principais:

Ailerons: são superfícies móveis instaladas nas pontas das asas no bordo de fuga e são acionadas simultaneamente nas duas asas e em sentidos opostos gerando uma diferença de sustentação nas asas fazendo com que a aeronave tenha um movimento de rolagem.

Profundor: superfície móvel instalada no estabilizador horizontal do conjunto de cauda que provoca um movimento de arfagem na aeronave, que por sua vez altera o ângulo de ataque da asa fazendo a aeronave subir ou descer. Em conjunto com o aileron faz com que a aeronave faça curvas com um menor raio.

Leme: Superfície móvel instalada no estabilizador vertical do conjunto de cauda; a sua deflexão provoca um movimento de guinada na aeronave fazendo com que a mesma gire em torno de si no eixo Z, conforme Figura 17.

Flap: são estruturas móveis instaladas no bordo de fuga da raiz da asa que alteram consideravelmente o perfil aerodinâmico da asa, gerando assim uma maior sustentação para uma mesma velocidade, por consequência aumentando o arrasto. Normalmente estas estruturas só são acionadas para pouso, decolagem ou em circunstâncias onde se faz necessário o aumento da sustentação ou do arrasto com objetivo de frear a aeronave.

3.2.2 Movimentos de controle de um Avião

Quando se estabilizam as forças de ação e reação de um avião para que ele se mantenha em voo alinhado com o horizonte, pode-se levá-lo a qualquer ponto controlando apenas os movimentos em relação ao seu centro de gravidade, conforme a Figura 17.

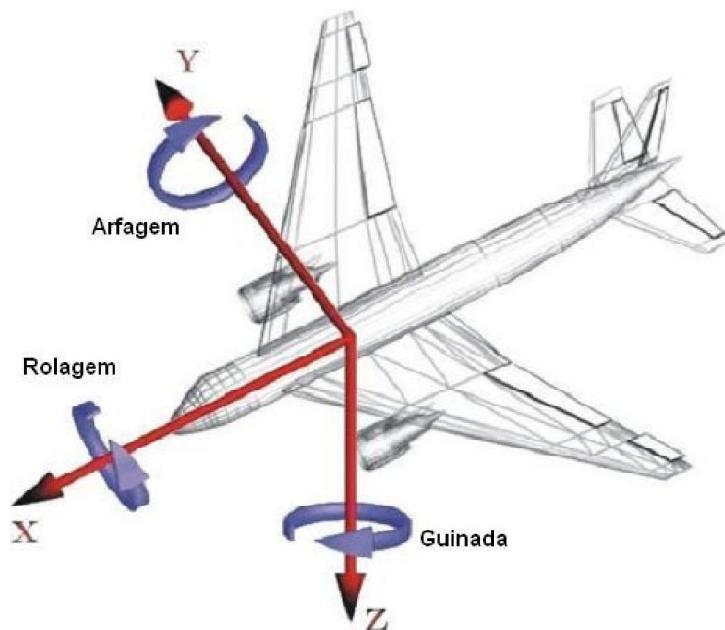


Figura 17: Movimentos de Controle em Relação ao Centro de Massa – Fonte: (ARAÚJO., 2007)

Arfagem: é o movimento em torno do centro de gravidade no eixo Y resultante da deflexão do profundor no grupo de cauda.

Rolagem: é o movimento em torno do centro de gravidade no eixo X resultante da deflexão oposta dos ailerons na ponta das asas.

Guinada: movimento em torno do centro de gravidade no eixo Z, resultante da deflexão do leme no grupo de cauda.

Todos os movimentos do modelo em voo são resultantes das somas das componentes resultantes da variação de posicionamento das estruturas móveis.

4 PROJETO E IMPLEMENTAÇÃO DA PLACA CONTROLADORA

Este capítulo descreve a construção de uma placa com uma versão inicial para testes e aperfeiçoamento e uma versão final com todas as correções necessárias ao funcionamento para o propósito específico de monitoramento de linhas de transmissão.

4.1 Sensores

Os principais sensores utilizados no projeto da placa são o GPS para o posicionamento do Vant, o magnetômetro para indicação de direção do modelo, o giroscópio e o acelerômetro, que são utilizados para fornecerem medidas de inclinação do modelo para controle de direção e estabilização.

4.1.1 GPS

De acordo com (SOARES DE LIMA, 2005) o Sistema de Posicionamento Global (*Global Positioning System* - GPS) foi inicialmente concebido pelo Departamento de Defesa (*Department of Defense* - DoD) norte-americano para o apoio à navegação. A Figura 18 indica de forma simplificada o aspecto de uma constelação de satélites semelhante à configuração que atualmente forma o segmento espacial do GPS e que busca o envio de sinais para toda a superfície terrestre.



Figura 18: Constelação de satélites (Fonte: Portal Vaz Tolentino)

O receptor de GPS capta o envio destes sinais dos satélites visíveis, uma vez que as frequências de transmissão e níveis de sinais não permitiriam a recepção de outros satélites fora da visada direta, entre as antenas transmissoras e receptoras.

Por um processo de triangulação o receptor analisa as informações de posição espacial enviada por cada satélite, as compara com os tempos de atraso de chegada de cada um dos satélites captados, obtendo assim uma coordenada geodésica do receptor em relação à constelação de satélites.

O tempo de atualização da informação de localização básico é de 1 s, mas pode ser alterado para mais ou para menos dependendo da qualidade e compatibilidade do receptor.

Receptores Genéricos de GPS normalmente possuem interface gráfica para interação com o usuário, onde pode-se armazenar rotas, pontos de passagem, traçar caminhos, dentre outras funcionalidades. Alguns possuem portas de comunicação USB, RS232, SPI para exportar os dados em tempo real ou atualização de mapas.

Módulos receptores GPS compreendem um sistema similar aos receptores genéricos com interfaces gráficas, mas podem atender a propósitos específicos, tais como determinar a trajetória percorrida por um sistema de monitoramento veicular e determinar a localização geodésica de sensores de demarcação ou levantamento topográfico. Esses módulos se conectam por interfaces como RS232, USB, SPI, Bluetooth. Para os sistemas de GPS esta comunicação deve seguir um protocolo.

Neste trabalho utilizou-se um modelo GPS de tamanho miniaturizado, Figura 19, com saída RS232 com níveis de sinal de 5 V. A comunicação com um módulo pode ser com protocolo proprietário ou com protocolo definido pela National Marine Electronics Association (NMEA), especificação 0183 ou comumente denominado de NMEA 183.

O protocolo NMEA 183 estabelece toda a parametrização física e elétrica da interface, tais como velocidade de comunicação, níveis de tensão e todos os tipos de informação que o GPS deve fornecer como latitude, longitude, velocidade, direção, sendo todas as informações enviadas periodicamente por sentenças. Os tipos de sentenças são selecionados na configuração do GPS de maneira a se reduzir o processamento com decodificação de dados não utilizados.

Estas sentenças têm um padrão de dados definido no protocolo, de maneira que, se necessário alguma informação adicional que não conste na sentença, deve-se configurar o GPS para emitir a sentença que contenha os dados necessários.

Apenas 2 tipos de sentenças são habilitados no receptor GPS NMEA: GPGLA NMEA – GPRMC. As sentenças são mostradas em Anexo 9.1.

Neste trabalho estas sentenças são configuradas para atualizarem os dados a cada 200 ms, com comunicação por uma porta serial a 115 kbps, sendo que esta configuração é feita pela porta serial do módulo e é armazenada em memória não volátil no próprio módulo. Desta forma só é necessário configurar apenas uma vez.

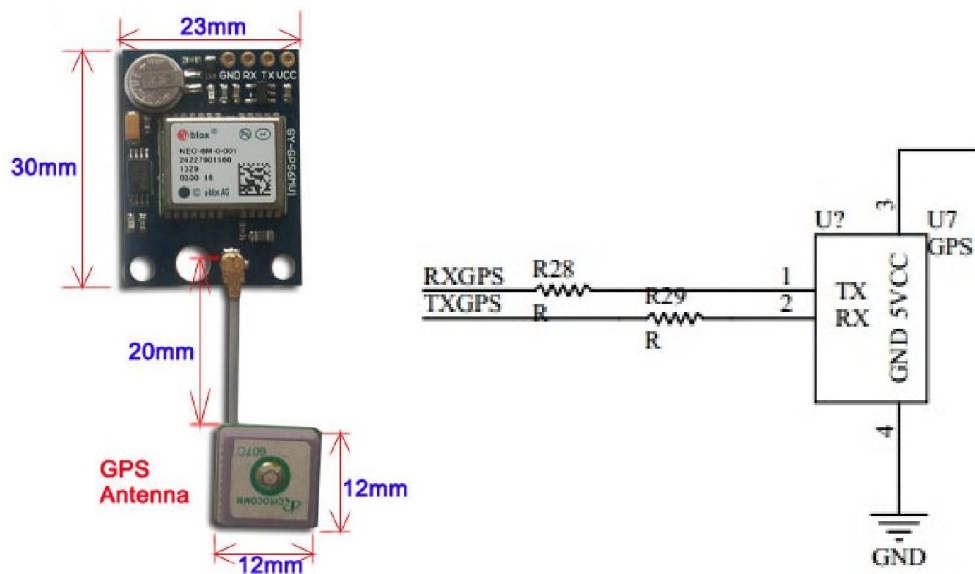


Figura 19: Módulo GPS Ublox 6m e Atena (Fonte: Aliexpress)

4.1.2 Magnetômetro

O sensor magnético utilizado na placa é o HMC5883, um sensor que contém elementos magneto-resistivos para a detecção do campo magnético terrestre, e tem sensibilidade de 8 Gauss e resolução de 5 miligauss.

O HMC5883 é um sensor tri-axial, possui toda lógica de controle e condicionamento dos sinais analógicos internamente ao invólucro, o sinal analógico é convertido em digital por um ADC de 12 bits, e pode fornecer até 160 amostras por segundo de cada eixo. O sensor possui interface I2C para configuração por meio de registradores internos e saída de dados por meio da leitura dos registradores.

Este sensor, por conter elementos magneto resistivos, e por não ser possível a fabricação de sensores química e fisicamente iguais, as leituras dos eixos podem apresentar variações de offset e amplitude para uma mesma intensidade de campo para cada eixo, sendo necessária a calibragem do sensor.

Por ser muito sensível, qualquer estrutura magnética que for posicionada próxima pode afetar a calibragem do sensor, sendo necessária uma nova calibragem para o correto funcionamento conforme pode ser observado na Figura 20. Ela mostra o procedimento de calibragem do magnetômetro usado no vant.

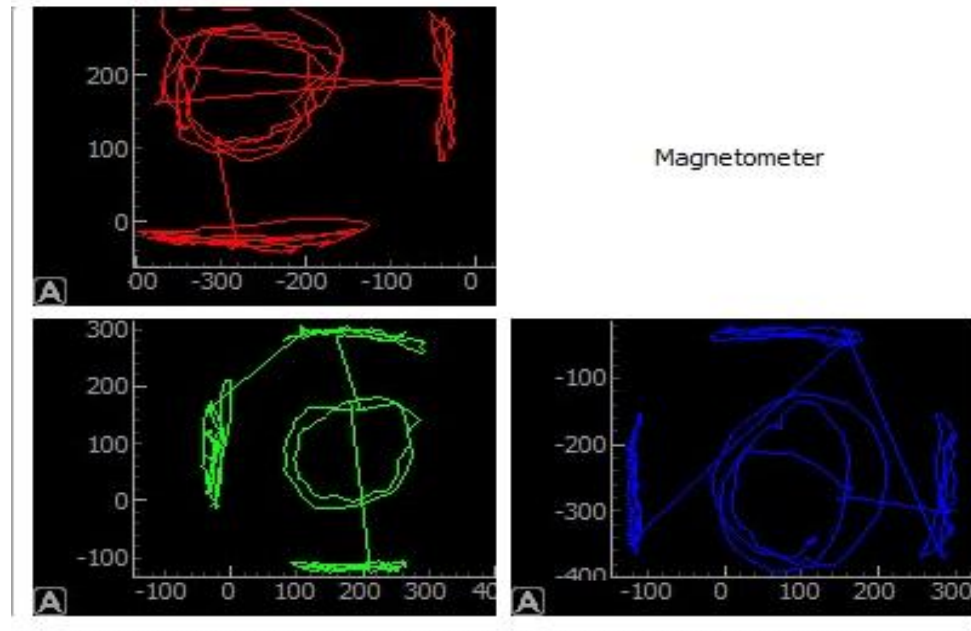


Figura 20 Erro offset ocasionado por proximidade de campo magnético

Observa-se que com o movimento circular em torno do próprio eixo, em um mesmo plano e sentido há um deslocamento do erro de offset entre os eixos XYZ decorrente da proximidade de uma estrutura ferromagnética.

Como a aplicação deste sensor na placa é a leitura da direção da placa, então torna-se necessário calibrar o sensor no ambiente de uso, sendo que primeiramente monta-se toda a estrutura na qual o sensor estará exposto; posterior a isto o sensor deve ser continuamente exposto ao campo magnético terrestre. Assim, estando distante de qualquer estrutura que contenha elementos ferromagnéticos, o sensor deve ser girado em várias posições, de preferência em passos de 30°, em que todos os eixos tenham trocado entre si os alinhamentos em relação ao norte magnético. Esta operação deve ser executada por completo por no mínimo 20 vezes para a obtenção de uma calibragem que resulte em leituras com precisão.

Durante a calibragem são mantidos os máximos e mínimos de cada eixo do sensor, desta forma pode-se obter o erro de offset de cada eixo, pela equação 4-1.

$$offset = (média\ max - média\ min)/2 \quad [4-1]$$

No caso das linhas de transmissão, além de sua função principal de conduzir energia elétrica, também são fontes geradoras de campo eletromagnético, campo este que pode alterar as medidas obtidas pelos sensores magnéticos, conforme se pode observar na Figura 21.

A figura apresenta resultados de campo para 3 eixos xyz quando o magnetômetro for posicionado a aproximadamente 8 m de uma rede distribuição de 13,8 kV.

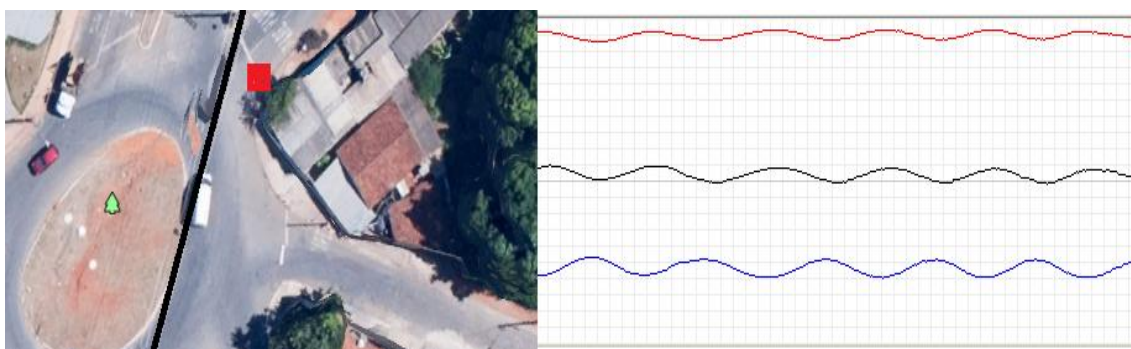


Figura 21: Sensor magnetometro posicionado a 8 metros da linha de transmissão.

A Figura 22 mostra que o sensor não capta qualquer perturbação da rede de distribuição a uma distância de 70 m da rede. Também mostra que para uma distância de 35 m a perturbação é pequena, porém já é perceptível.

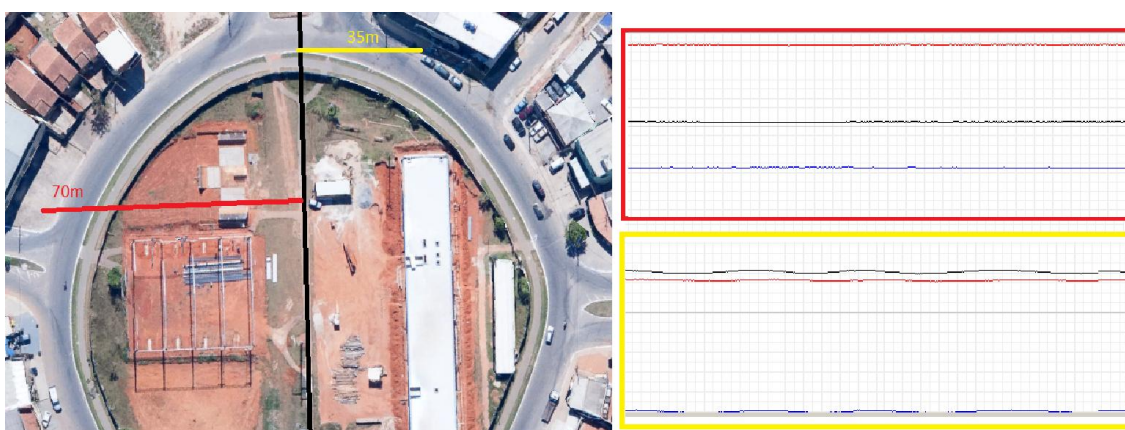


Figura 22 Medidas do sensor a 70 m e 35 m de distância da linha de distribuição.

Estas perturbações provocadas pela rede são diretamente proporcionais à corrente que circula na linha de transmissão. O controlador deve ser capaz de eliminar essas perturbações, pois a resultante deve ser apenas a indicação de orientação em relação ao norte magnético.

Torna-se prudente a utilização do sistema a uma distância mínima de 35 m de linhas de transmissão e redes de distribuição, tanto para evitar ruídos no sistema de rádio controle que poderia derrubar o modelo por erro de comando do piloto, bem como evitar erro de direcionamento devido à leitura do sensor magnético. Caso o controlador não tenha um processo para eliminar tal perturbação, poderá perder a função.

Com a aproximação de uma rede elétrica, o sensor sofre interferência do campo eletromagnético da rede, o que deforma a leitura do sensor. Esse erro, caso o magnetômetro seja usado na orientação do modelo, tornaria a navegação próxima a linhas de transmissão impraticável. Para corrigir esta distorção é aplicado um filtro de média móvel das últimas 32 leituras.

Mesmo sendo um filtro muito simples, é suficiente para se retirar praticamente toda interferência do campo eletromagnético da rede. A Figura 23 apresenta os resultados da atuação do filtro implementado para reduzir a influência dos campos magnéticos de redes elétricas no magnetômetro.

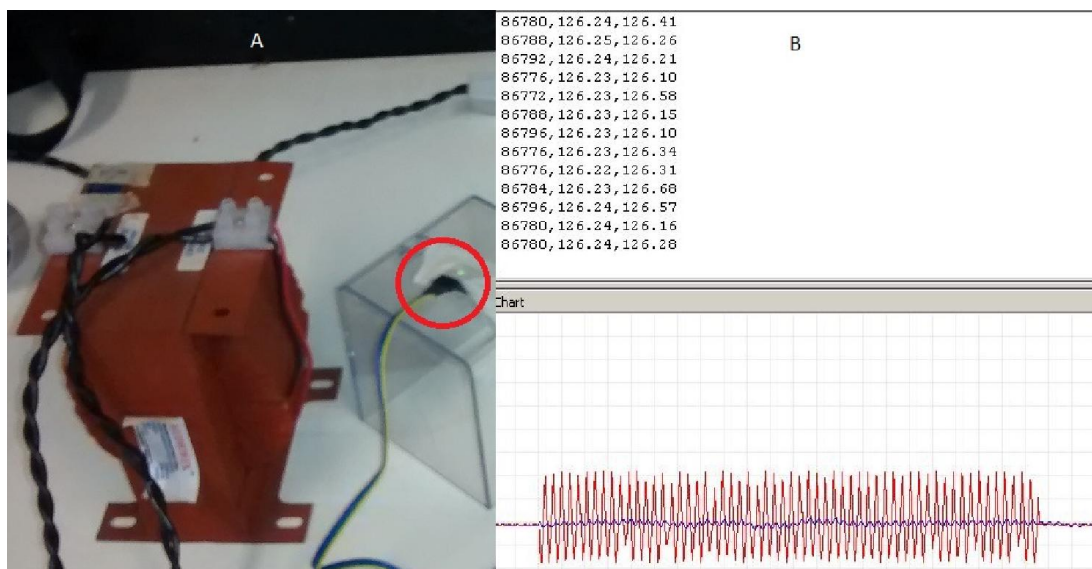


Figura 23 (a): Sensor magnético HMC5883 ao lado de trafo 1 kVA. (b): filtragem

O sensor foi montado próximo a um transformador de 1 kVA operando a vazio. No gráfico de resultados, Figura 23b, o traço vermelho representa a saída do sensor sem a

presença do filtro; a linha azul representa a saída do sensor após passar pelo filtro implementado. O resultado mostra que a resposta do filtro atenua consideravelmente o sinal de campo magnético do transformador, ou seja, a saída corresponde apenas ao campo magnético da Terra.

Para o correto funcionamento este filtro deve ser aplicado diretamente nas saídas dos conversores de cada canal antes da conversão dos dados em ângulos.

Deve-se perceber que com a aplicação do filtro o tempo de resposta do sensor também é alterado, uma vez que no filtro de média móvel sua banda passante é uma função do tamanho da janela da média-móvel. A representação discreta de um filtro do tipo média-móvel pode ser vista na equação 4-2 (BORGES, 2015).

$$y[k] = \frac{1}{N+1} \sum_{k=0}^N x[n - k] \quad [4-2]$$

Onde:

$Y[k]$ =valor de saída do filtro

$N+1$ = Número de amostras utilizadas

$x[n-K]$ = Conjunto de amostras a serem somadas

Na Figura 24 observa-se a conexão da placa de sensores interligada ao barramento I2C da placa controladora. Nesta placa estão presentes os sensores de pressão barométrica, acelerômetro, magnetômetro e giroscópio. A substituição dos sensores discretos por um conjunto em uma placa discreta é decorrente da possibilidade de evolução do atual projeto facilitando assim o trabalho de novos pesquisadores.

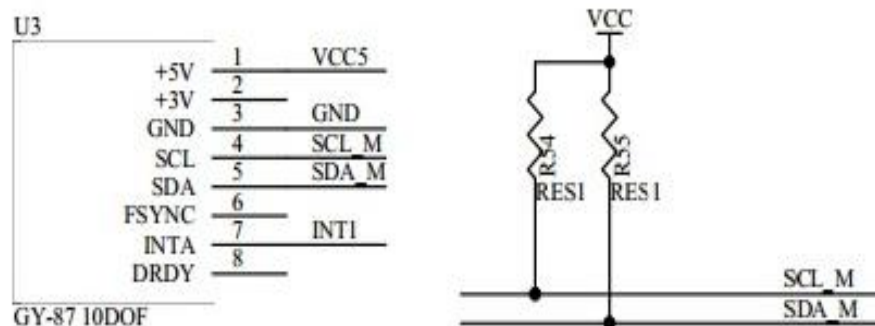


Figura 24 Conexão da nova placa de sensores ao barramento I2C

4.1.3 Giroscópio Acelerômetro

Dentre os sensores utilizados neste trabalho os de maior relevância para o funcionamento correto do sistema são os sensores de aceleração e o giroscópio. É a partir dos sinais desses dois sensores que são obtidos os dados de correção em relação ao ângulo de inclinação necessários tanto para estabilização como direção a ser seguida.

O giroscópio utilizado neste trabalho é o MPU6050 (compatível com MPU9250 exceto por não incluir internamente o magnetômetro) da Invesense. Este é um componente com dois sensores MEMs internos: um giroscópio e um acelerômetro, ambos com 3 eixos cada. Estes sensores podem ser acessados e controlados independentemente um do outro.

O MPU6050 possui três conversores AD de 16 bits para digitalizar os três eixos simultaneamente do giroscópio e mais três conversores AD de 16 bits para digitalizar simultaneamente os três eixos do acelerômetro. Para precisão na detecção de movimentos lentos o sensor tem faixas de sensibilidade para cada um dos sensores que devem ser habilitadas de acordo com o propósito de uso (INVENCENCE, 2013).

A disposição de orientação dos três eixos do sensor pode ser observada na Figura 25 bem como a sua pinagem e dimensão do invólucro.

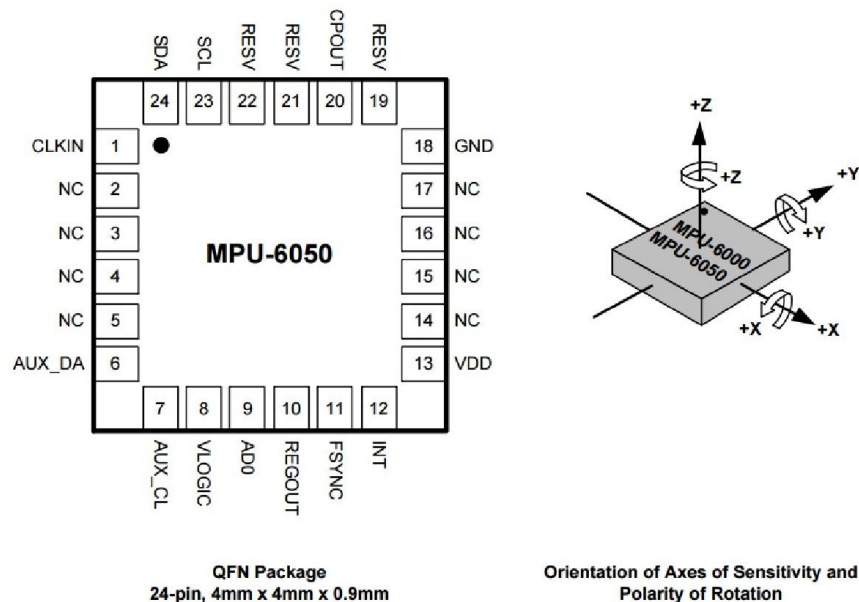


Figura 25: Sensor Giroscópio e Acelerômetro (fonte: (INVENCENCE, 2013))

A comunicação do sensor com o mundo exterior é feita por um barramento I2C com frequência de até 400 kHz. Através deste barramento pode-se obter os dados de todos os registradores internos do sensor bem como configurá-lo para o propósito de uso.

O MPU6050 possui um *Buffer* FIFO de 1024 bytes para armazenar os dados dos sensores. Este *buffer* pode ser utilizado para reduzir o consumo de energia e uso do barramento, já que o sensor pode efetuar leituras periódicas e armazenar no buffer para que o mesmo seja descarregado em intervalos de tempo maiores (INVENCENCE, 2013).

O MPU6050 possui ainda uma porta auxiliar do barramento I2C por onde o sensor pode ser conectado a um magnetômetro para a fusão interna dos dados dos sensores e formar um sistema de 9 graus de liberdade.

Para aplicações em que o processamento seja reduzido, o sensor possui um Processador Digital de Movimentos (DMP). Este processador pode tratar toda parte de cálculo dos sensores fornecendo apenas os dados de saída já tratados, como ângulos de inclinação e direção, reduzindo assim a dependência de processadores matemáticos para trabalhar em tempo real. O DMP tem de ser programado tal qual um microcontrolador, o sensor na inicialização do sistema deve ser programado juntamente com o DMP de acordo com a aplicação do sensor.

A fusão dos dados dos sensores não é o foco deste trabalho, os sinais dos sensores utilizados para estabilização neste trabalho são tratados diretamente no DMP e o código do mesmo é fornecido pelo próprio fabricante em código de máquina, sendo omitido qual núcleo e set de instruções, portanto não há a possibilidade de interpretação sobre o código.

Existem diversos grupos de aplicação e desenvolvimento no sentido de tratamento e filtragens de ruído relativos à utilização de sensores de baixo custo aplicados a VANTS; um bom ponto de partida voltado ao tratamento e fusão dos dados extraídos dos sensores inerciais, é retratado em (OLLIW, 2013), em que é feito um breve comparativo sobre filtro de Kalman, filtro complementar e filtro de Mahony.

4.2 Circuito Elétrico da Placa

A atuação da placa controladora sobre as superfícies de controle e sobre o motor responsável pela estabilização e propulsão depende de informações sobre a situação atual do modelo.

Assim, alguns sensores são essenciais ao sistema de controle. São sensores que podem determinar a orientação do avião no espaço, sua altitude, sua velocidade e sua posição geográfica.

Essas informações são usadas para realimentar o sistema de controle e garantir que o modelo siga as informações de referência. Para tanto o circuito é baseado em um processador central e vários sensores de entrada e atuadores de saída, conforme diagrama de blocos da Figura 26.

Levando-se em consideração as possíveis influências eletromagnéticas decorrentes da aproximação às linhas de transmissão de alta tensão, a placa é dotada de malha de aterramento e deverá ser acondicionada em estrutura compatível para a blindagem de possíveis interferências indesejadas.

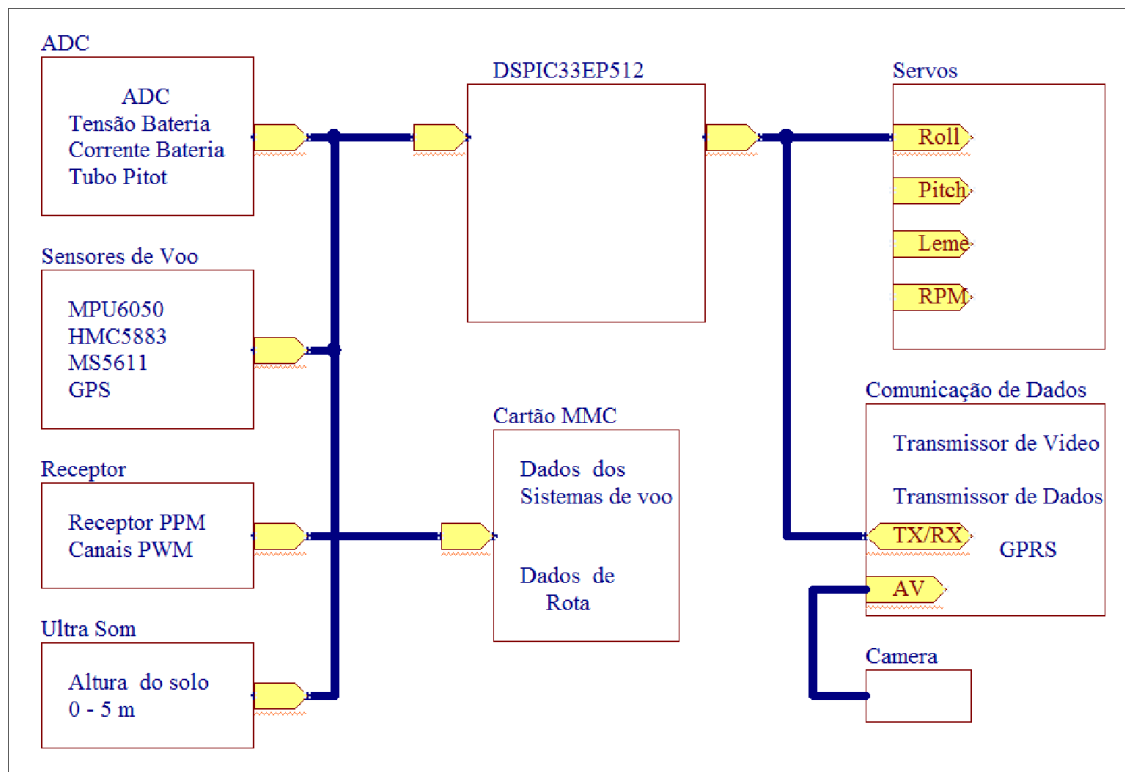


Figura 26 Diagrama de blocos do circuito.

O circuito da placa de circuito impresso foi desenvolvido no software Altium Designer, em um único circuito que contempla todos os componentes utilizados na placa, diferente da representação de circuitos de ligação unifilar em que a representação conecta

todos os pontos por linhas. Neste circuito o esquema de ligação é feito por NETs (ligações indicadas por nomes) conforme pode-se observar na Figura 27. No Anexo 9.6 o circuito é detalhado e pode ser observado em detalhes.

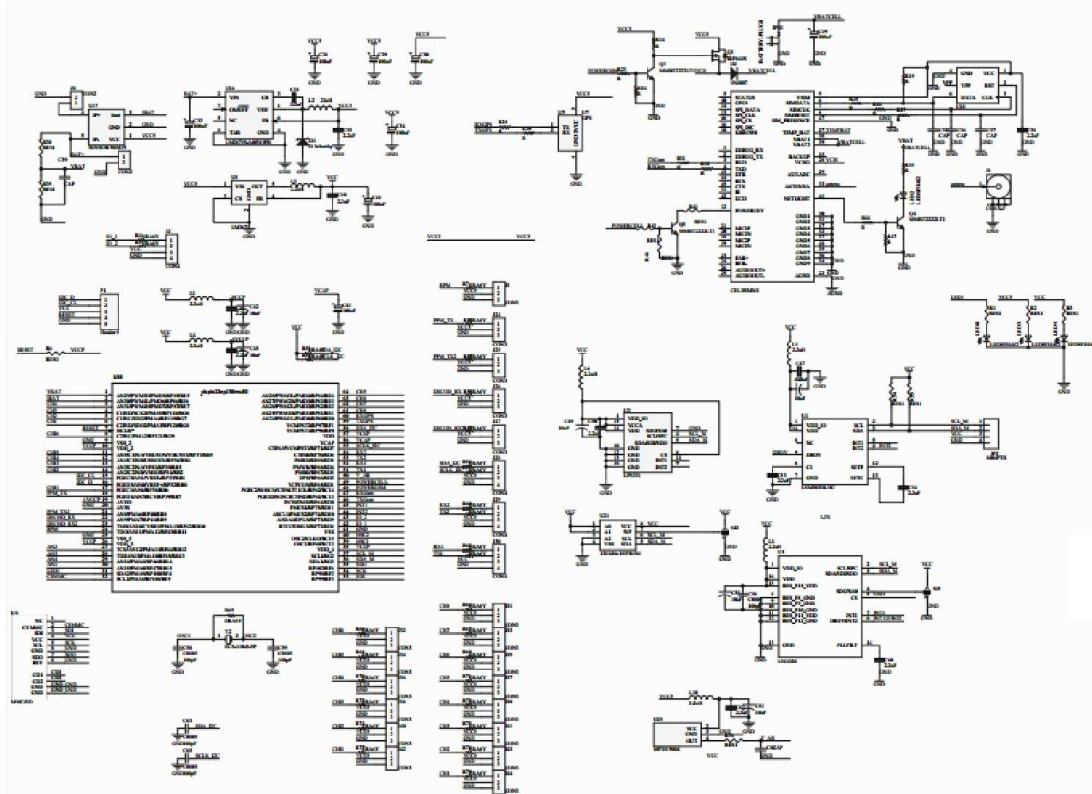


Figura 27 Esboço do circuito da placa controladora V1.0

Alguns sensores são pertinentes ao próprio funcionamento da placa, como sensor de temperatura interno ao sensor MPU6050.

O circuito da placa disponibiliza várias conexões para interfaceamento externo, e algumas funcionalidades internas, enumeradas a seguir:

- 3 portas seriais (GPS, rádio comunicador, GPRS)
- 1 porta I2C (Sensores, Memória)
- 2 entradas PPM (RX controle remoto)
- 1 saída PPM (Expansão de Servos)
- 6 canais decodificador PWM (entrada do receptor de controle)
- 8 saídas para atuadores (controle dos servos).
- 1 sensor de corrente até 50 A (corrente da bateria)
- Monitor de tensão de bateria (Controle de carga)

- 1 porta de interfaceamento para sensor de proximidade do solo (aterrisagem)
- 1 sensor de pressão diferencial para leitura da velocidade do ar (sustentação)
- Fonte chaveada de 5V, 5A (alimentação dos servos)
- 4 canais AD 12 bits para propósitos diversos.
- 1 módulo de comunicação GSM (redundância de comunicação)

No Anexo 9.6 pode-se observar com riqueza de detalhes o circuito separado por blocos, bem como a imagem 3D da placa roteada na versão 1.2. A Figura 27 apresenta um esquemático do circuito por completo da versão 1.0. A principal diferença entre as versões é a troca dos sensores antes por soldagem direta na placa e a nova versão com todos sensores inerciais em um circuito dedicado soldado sobre a placa. A troca dos sensores foi motivada pela dificuldade de soldagem em comparação ao circuito dedicado.

4.3 Placa Controladora em Circuito Impresso

Para aplicação dos recursos necessários na elaboração da placa optou-se por manter todo o circuito em uma placa, de maneira a evitar ruídos e problemas de conexão. Um esboço da placa em 3D pode ser observado na Figura 28.

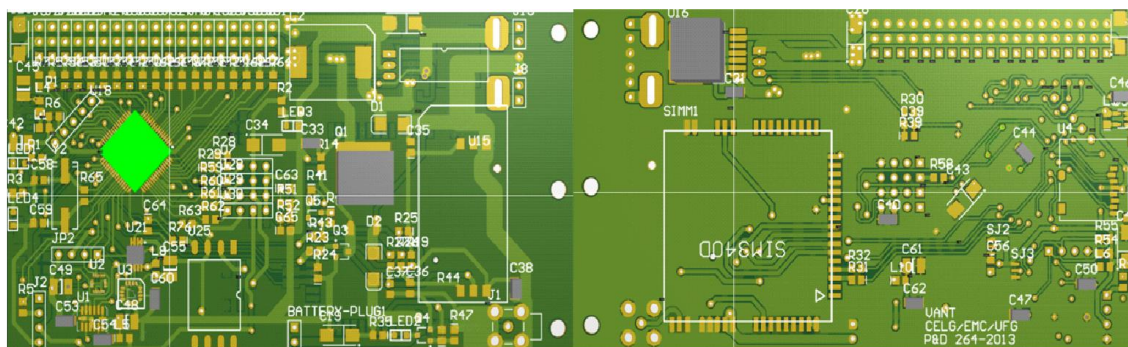
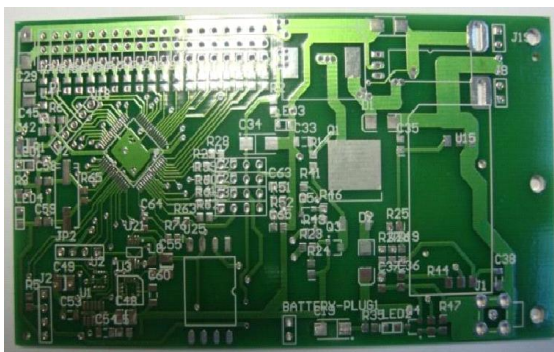
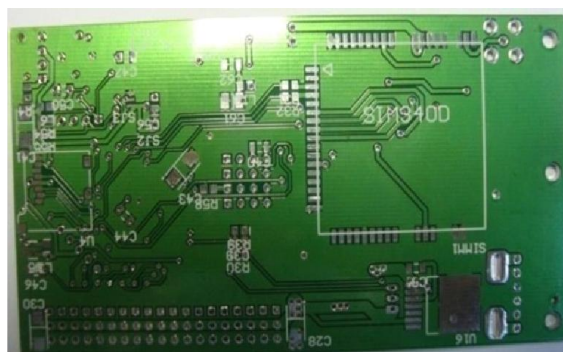


Figura 28 layout 3D superior e inferior V1.0

Na Figura 29 pode-se observar a fidelidade entre a projeção 3D e a placa já confeccionada. A Figura 30 apresenta o circuito da placa controladora desenvolvida já montada com os principais componentes



Layout Superior



Layout Inferior

Figura 29 Placa já confeccionada V1.0.

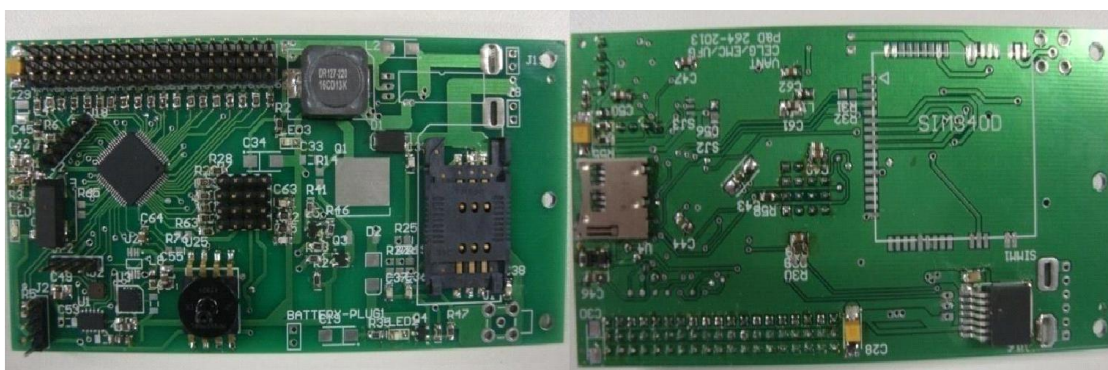


Figura 30 Placa montada

5 SOFTWARE DE CONTROLE

Neste capítulo é abordado o software da placa controladora e todos os subsistemas necessários para o correto funcionamento do sistema, e o sistema de monitoramento em solo denominado Software de Programação e Acompanhamento de Missão (SPAM).

5.1 Reset e inicialização

Após o *reset* físico do DSP o firmware configura os ponteiros destinados ao ambiente de execução decorrente da utilização da linguagem C, tais como *heap*, *stack* e declarações de constantes. É feita uma verificação nos registros internos de qual o motivo de reset e é feito um tratamento específico caso seja detectado endereçamento errado por ponteiros, já que o DSP detecta o erro quando algum registrador aponta para algum endereço inexistente na memória física do mesmo.

Após o reset do sistema o fluxo do programa é desviado para a região final da memória de programa, onde é feita uma verificação por checksum da parte inicial desta mesma memória para validar se o DSP está programado corretamente, com isso é possível antes de qualquer voo manter a redundância do programa. Caso a memória esteja programada de forma correta, com valor de inicialização diferente de 0xffff, o sistema faz uma verificação de uma porta de entrada e identifica se está habilitado para a atualização do programa.

Se o sistema identificar a liberação da chave então ele configura as portas seriais e aguarda os comandos do SPAM para a inserção de um novo programa sobrescrevendo o antigo. Este processo é denominado de *bootloader*, e não são todos os DSPs que podem efetuar o *bootloader*, isso porque o mesmo deve ser dotado de funções específicas como conter instruções que permitam que ele mesmo faça leituras e escritas na própria memória de programa, e deve conter uma região na qual possa gravar um código que interfaceie todos os comandos de leitura e escrita na memória de programa.

Uma vez que o sistema inicializa o *bootloader*, não mais poderá reverter o processo, pois a primeira função executada é apagar o programa antigo.

O DSP em uso tem uma região de memória específica que deve ser habilitada para funcionar corretamente, denominada *Auxiliary Program Flash Memory*. A Figura 31 apresenta o mapa de memória do DSP com faixa de memória separada. É nesta faixa de memória que se grava o software de *bootloader* que deve ser protegida do restante. Assim o

bootloader não seria apagado acidentalmente. O software do *bootloader* pode utilizar de qualquer meio de comunicação disponível, como I2C, SPI, cartão de memória, porta serial dentre as várias possibilidades para regravar o *firmware*.

Neste projeto utilizou-se da porta serial como meio de transferência para o *bootloader* tendo em vista que toda comunicação do sistema se baseia na mesma. Para tanto foram inseridos novos serviços de protocolo de comunicação, o qual será retratado no item de comunicação.

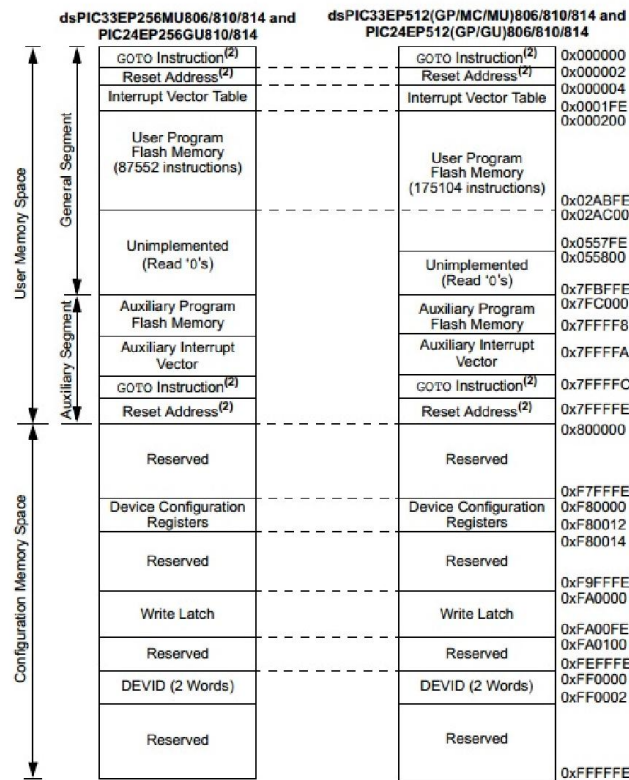


Figura 31 Mapa de memória do DSPIC33EP512 (MICROCHIP T. , 2011)

Tendo como base a nota de aplicação “dsPIC33e/PIC24E Bootloader” do próprio fabricante do DSP (MICROCHIP, 2011), o código da interface de gravação utilizado no computador foi inserido no SPAM, assim foi necessário apenas readequar o protocolo utilizado na nota de aplicação e inserir um botão de gravação no SPAM, para que qualquer alteração no programa da placa seja transferido para o modelo, sendo necessário apenas que o mesmo esteja parado e com a chave de gravação habilitada.

Após o sistema efetuar as configurações de *reset* e verificar o *bootloader*, o sistema passa então a configurar as interfaces periféricas sendo que a primeira deve ser o PLL da CPU

para frequência de trabalho em 80 MHz ou 40 MIPS, habilita temporizadores para interrupção de 1 ms, habilita o temporizador 4 e 5, configura as portas seriais para comunicação em 115200 kbps e habilita a interrupção de recepção de cada uma. E ainda habilita a porta SPI utilizada no cartão de memória para 10 MHz, configura 2 entradas IC para decodificação do sinal PPM do receptor e as associa ao temporizador 4; configura 6 entradas para decodificação para sinais PWM do receptor e as associa ao temporizador 4; configura as saídas dos canais de controle e associa ao temporizador 5; habilita 5 canais do ADC para 12 bit e as associa às portas de entrada. Finalizando, remapeia os pinos do DSP de acordo com PCB do circuito. A configuração de funcionamento de um DSP sempre dificulta o início do trabalho de programação propriamente dito, assim o trecho do programa pertinente à configuração foi inserido no Anexo 9.89.8 no intuito de auxiliar a compreensão.

5.2 Fluxo do programa

Tendo em vista que o sistema como um todo tem apenas uma unidade de processamento que concentra todas as informações dos sensores e atuadores, qualquer bloqueio de funcionamento pode resultar em perda total do modelo com risco para os operadores. Visando a segurança, o software usa o princípio de execução baseado em eventos. A Figura 32 apresenta um fluxograma simplificado do modo de trabalho do sistema.

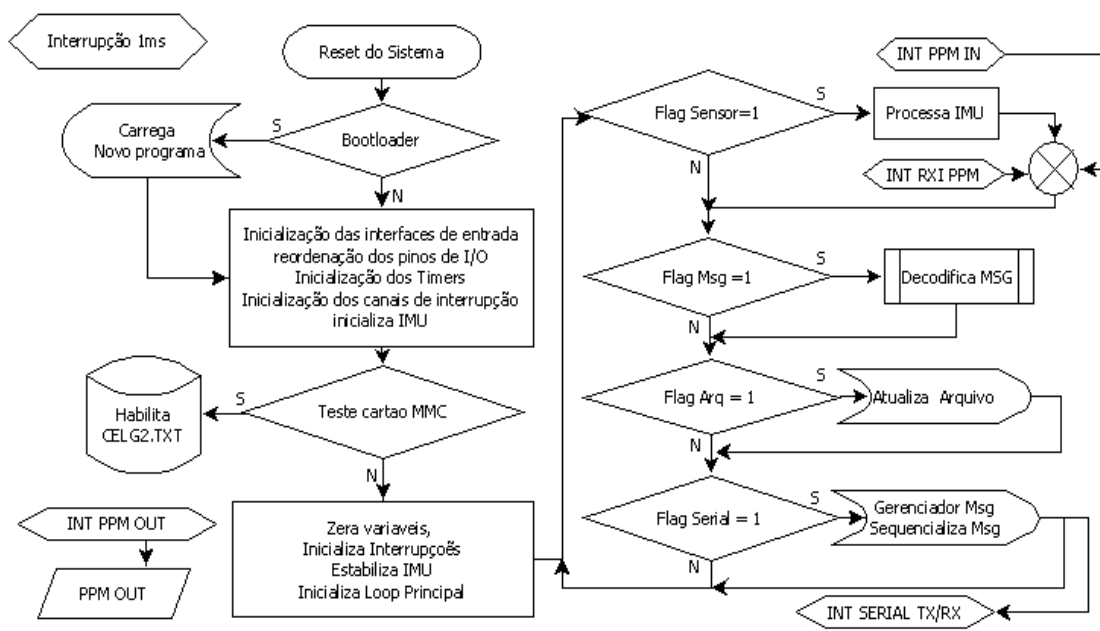


Figura 32 Fluxograma do software

A rotina principal concentra o foco de controle chamando as funções auxiliares de acordo com as *flags* setadas nas chamadas de eventos, sincronizando assim todos os eventos como leitura dos dados dos sensores, envio de mensagens e atualização dos dados de saída.

Além de chamar funções auxiliares na rotina principal, é montada a lógica do controle. Nesta parte do programa são executados os comandos de coordenação da rota traçada e o armazenamento dos dados de voo.

Esta rotina fica em um *loop* fechado verificando as *flags* de eventos. Estas alterações de *status* das *flags* decorrem da temporização interna da interrupção ou por contagem de eventos. Nos eventos decorrentes de interrupção o foco do programa é desviado para a rotina específica de tratamento, por exemplo, a que recebe os dados do GPS.

5.3 Interrupção

O DSP utilizado possui uma grande variedade e modos de interrupção, que é utilizado também para manter a estabilidade do sistema. Utilizando as interrupções como eventos podemos reduzir o foco da rotina principal reduzindo as chamadas de entrada e saída; para tanto foram habilitadas as interrupções dos temporizadores, IC, OC e portas seriais como explicado nas subseções a seguir.

5.3.1 Temporizadores

Apenas o *Timer1* utiliza a interrupção, que é configurada para interromper o sistema a cada 1 ms. Interno à rotina de tratamento, é feita a marcação dos eventos que são tratados por tempo. A exemplo dos sensores inerciais, a rotina completa pode ser observada no anexo 9.2. Interno à interrupção do *Timer1*, também é tratado o *buffer* de recepção da porta serial, conectada ao GPS e/ou rádio; esta porta passa por uma verificação a cada 2 ms para saber se chegou algum pacote de dados do GPS e/ou rádio. Isso é feito pelo fato de se utilizar a FIFO de 4 bytes para interrupção da porta serial. Assim, a mesma só pede interrupção ao sistema se preencher por completo, o que poderia ocasionar a perda dos pacotes não múltiplos de 4 bytes.

As marcações de eventos, tais como atualização dos sensores e controlador PID, também são feitas na interrupção do *Timer1*, e ambas são sincronizadas em tempos iguais de 25 ms.

5.3.2 Input Capture

As entradas de sinais de controle dos servos, oriundas do controle remoto, utilizam um canal de Input Capture (IC) cada. São sinais periódicos e a informação é contida no nível lógico 1, que pode variar de 1 a 2 ms, representando a proporção do curso total do servo.

As duas entradas de sinais PPM oriundos do receptor de controle remoto são decodificadas internas às interrupções de cada canal IC correspondentes à entrada. Estes sinais têm como característica a posição dos pulsos, considerando que apenas a largura do pulso em nível lógico zero varia (padrão Futaba). O pulso em nível lógico 1 tem tamanho fixo e representa a marcação de contagem para um novo canal. A periodicidade de atualização dos canais é de 20 ms e é reiniciada pela ausência de sinais por um período de 8 ms, sincronizando assim a atualização dos canais. Na Figura 33 pode-se observar o sinal PPM para um sistema de 8 canais. Este sinal é analógico pois a informação de cada canal contida nele é uma variação do tempo em nível lógico 0. Para tanto, o *firmware* faz um tratamento de erro considerando tempos mínimos e máximos para os níveis lógicos 1 e 0. Caso o pacote seja considerado inválido o mesmo é descartado,

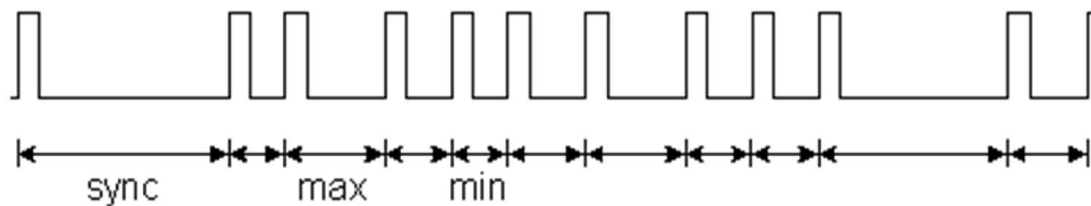


Figura 33 Sinal Serial PPM (Fonte: Portal Futaba)

5.3.3 Output Compare

Os sinais de saída que controlam os servos são similares aos sinais de recepção dos servos, com periodicidade de 20 ms e a informação é contida no nível lógico 1 que pode variar de 1 a 2 ms, representando a proporção total do curso do servo. Cada canal *Output Compare* gera o sinal de cada servo de forma independente. A Figura 34 apresenta o padrão de sinais de controle do servo.

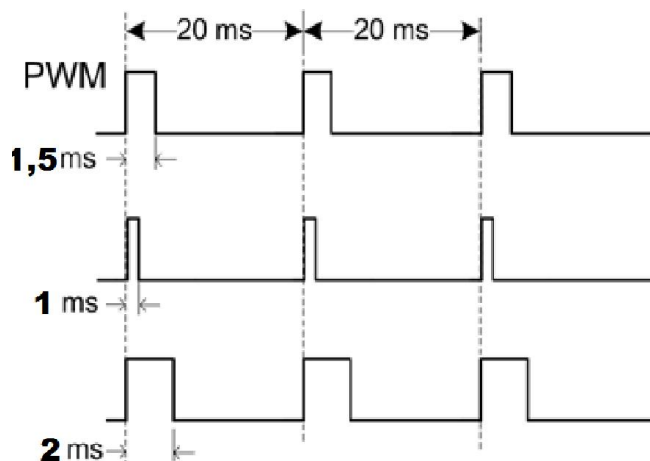


Figura 34 Sinal analógico para controle de servo. (Fonte Junior, Siqueira)

5.3.4 USART Portas Seriais

Na recepção das portas seriais responsáveis pela comunicação com GPS e RX/TX as interrupções são habilitadas por preenchimento da FIFO de 4 bytes, e são auxiliadas pelo Timer1 para não haver perda de pacote. Assim o sistema tem um menor número de chamadas de interrupções por byte recebido.

Na transmissão o sistema descarrega toda informação a ser enviada, que é colocada em um *buffer* de memória, e habilita a interrupção por leitura da FIFO, onde a cada leitura completa da FIFO, a rotina de interrupção preenche novamente a FIFO com os dados do *buffer* até que toda informação seja enviada, reduzindo desta forma o número de interrupções por byte enviado.

5.4 Comunicação

A comunicação é feita quase em sua totalidade por dados seriais, com exceção do cartão de armazenamento, que também transfere dados em qualquer direção considerando que pode-se ler e escrever no cartão.

Para a comunicação através de dados seriais os mesmos devem obedecer a um padrão de comunicação, de modo a ser possível recuperar as informações e comprovar a validade da mesma. Para tanto, desenvolveu-se um protocolo específico para o sistema. Esse protocolo é descrito na seção seguinte.

5.5 Protocolo

O sistema usa a comunicação baseada em um protocolo que visa um maior alcance utilizando os rádios em uma topologia *mesh*, onde um ponto de comunicação pode ser apenas um retransmissor sem intervir na comunicação, isso devido a possibilidade de haver um Vant voando a uma distância além do alcance, e um segundo Vant servir de ponte para os dados.

Para implementação de um protocolo com topologia *mesh* é necessário que os dados contenham informações de origem, destino e tempo de vida, para tanto baseou-se o desenvolvimento no conceito do protocolo TCP/IP por sua robustez e capacidade de endereçamento.

No protocolo criado foi omitido o tempo de vida tomando se como base a rede mesh e substituído por tabelas internas de detecção de recebimento, assim todo ponto de comunicação pode retransmitir um pacote caso o mesmo não seja de seu endereçamento e o destinatário conte da lista. Para tanto é necessário que o pacote a ser retransmitido seja enviado na forma de serviço para retransmissão.

Considerando a utilização constante e visando a redução de perda de dados criou-se vários tipos de envios denominados serviços, assim pode-se quebrar a grande quantidade de informações e classificá-las quanto à periodicidade de envio. Não sendo necessário o protocolo não se exige confirmação de recebimento. Pode-se citar como exemplo os dados de estabilização instantânea do modelo para visualização no SPAM; este serviço é enviado com periodicidade configurável e não requer confirmação. Pode-se exemplificar também a gravação de configuração de estabilidade como os ganhos dos controladores, que são dados por demanda e requerem confirmação por parte do SPAM; caso não haja uma resposta de confirmação o pacote é retransmitido.

A divisão do protocolo por serviços permite classificar a condição de recebimento ou rejeição, para evitar acidentes ou alterações de configuração no modo autônomo, alteração de rota, a parametrização é favorecida quando se permitida a classificação dos dados.

O protocolo é basicamente constituído por:

STX start: obrigatoriamente o caractere 0x02 no início de qualquer pacote

Ds: endereço de destino (1 byte)

Og: endereço de origem (1 byte)

Sc: serviço transmitido (1 byte)

Dados: dados relativos ao serviço em questão, com tamanho variável de bytes

Ck: checksum checagem matemática do pacote completo (1 byte).

ETX: caractere de término do pacote obrigatoriamente 0x03.

Os caracteres especiais deste padrão são apenas 3: STX, ETX e ESC. STX e ETX são os delimitadores de início e fim de um novo pacote e, por conseguinte, não podem ser utilizados em nenhum momento, seja como dado ou endereço. Entretanto não se pode suprimir o uso de valores de dados nos valores internos aos dados utilizados no sistema. Para corrigir essa distorção adota-se o caractere de escape (*Byte stuffing*) que tem valor 0x7f, e em sequência adiciona-se 0x10 ao caractere que causaria confusão. Com isto informamos ao sistema que o mesmo deve remover o caractere de escape e subtrair o próximo byte de 0x10 para dar a sequência correta de dados recebidos.

Um exemplo de pacote para incluir um *waypoint* cujo serviço é 01, a latitude é 37.624924, longitude -122.389683, Altura de 100 m e índice 253, obtém-se a sequência de *tags* e dados de acordo com o protocolo: (STX, DS, OG, SC, float lat, float lon, int alt, int indice, CK, ETX) cujos valores em hexadecimal correspondem à sequência (02 01 80 01 EC 7F 8F 16 42 85 C7 F4 C2 64 00 FE 00 29 03). Na Tabela 2 é apresentada toda a sequência de valores correspondentes às tags e variáveis.

Tabela 2 Exemplo de pacote enviado ao modelo.

Variaveis e tags	Tipo de dado	Valor decimal	Valor em hexadecimal
STX start	char	02	0x02
DS destino	char	01	0x01
OG Origem	char	128	0x80
SC serviço	char	01	0x01
Latitude	float	37.624924	0xEC7F1642
Longitude	float	-122.389683	0x85C7F4C2
Altura	unsigend Int	100	0x6400
Indice	unsigned int	253	0xfe00
CK:checksum	char	41	0x29
ETX:stop	char	03	0x03

O *checksum* é um valor obtido por meio de uma função polinomial Anexo 9.7 e tem a finalidade exclusiva de confirmar a validade dos dados enviados. A função é repetida em todos os pontos de recepção e transmissão e a validade dos dados confirmada.

A quantidade de serviços está limitada a 255 serviços. Mesmo sendo relativamente pequeno é mais que suficiente para a aplicação. Até a versão atual (versão 1.0.5.1 do SPAM) foram implementados 17 serviços, sendo eles descritos conforme Tabela 3.

Tabela 3 Serviços implementados na versão 1.0.5.1 SPAM

Nº. do Serviço	Serviço
1	Serviço de gravação do waypoint SPAM → Modelo
2	Status do modelo Geral do modelo Modelo → SPAM
3	Status básico do modelo Modelo → SPAM
5	Confirmação de gravação Waypoint Modelo → SPAM
6	Resposta de leitura Waypoint Modelo → SPAM
8	Retorna ao ponto de origem SPAM → MODELO
10	Status Horizonte Modelo → SPAM
20	Ler Memória EEP
23	Gravar Memória EEP
29	Reset para bootloader
30	READ_BOOT_INFO BOOTLOADER
31	ERASE_FLASH BOOTLOADER
32	PROGRAM_FLASH BOOTLOADER
33	READ_CRC BOOTLOADER
34	JMP_TO_APP BOOTLOADER
40	BOOTLOADER sai da rotina do programa para o bootloader

5.6 Controladores

Após gerar uma rota no SPAM e transferir para o vant, o sistema inicia o processo de condução do modelo pelos pontos demarcados na referida rota. Para tanto, todo o sistema deve estar em operação e algumas condições são imprescindíveis para garantir a segurança de voo, como o sistema de GPS que deve ter dados de no mínimo 4 satélites e as condições climáticas favoráveis ao voo controlado.

Para o perfeito controle do modelo, são implementados controladores em malha fechada; estes controladores funcionam independentes entre si e utilizam como referência os

dados da rota programada, juntamente com as leituras dos sensores conforme esboço da Figura 35.

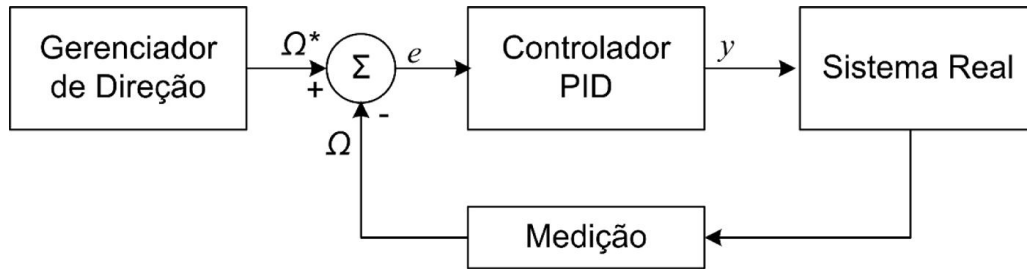


Figura 35 Modelo de controle utilizado

Para manter o controle do modelo dentro dos limites de voo, os dados de posição do próximo ponto de passagem e da posição atual do modelo são enviados a uma função denominada gerenciador de direção.

O gerenciador de direção utiliza as latitudes e longitudes do último pacote de dados do satélite e o próximo ponto de passagem. Assim ele calcula a direção atual que o modelo deve tomar com base no deslocamento (o código da função encontra-se no Anexo 9.3). A direção é então enviada aos controladores de estabilidade do modelo.

Para cada eixo de controle do modelo existe um servo específico acionado por um controlador dedicado, que executa todas as ações através dos servos, de forma a zerar os erros de posição do modelo para a condição desejada.

O modelo de controlador adotado é o Proporcional-Integral-Derivativo (PID) que é um controlador vastamente utilizado por sua simplicidade de implementação e vasta biblioteca disponível.

O controlador PID é na realidade a soma dos controles proporcional, integral e derivativo em um único controlador, possibilitando assim um controle com maior eficácia para várias aplicações distintas, através da manipulação dos ganhos de cada controlador individualmente, de acordo com a necessidade de ação do controlador. A equação 5-1 representa o controlador PID, onde K_p é o ganho proporcional, K_i é o ganho integral e K_d é o ganho derivativo.

$$y = K_p * e + K_i * \int_0^t e * dt + K_d * \frac{de}{dt} \quad [5-1]$$

Conforme já explicado no item 3.2, um modelo para se manter em voo, deve ter suas forças de atuação equilibradas para um voo plano e em linha reta. Para atuarmos na direção e altitude deve-se atuar diretamente nas estruturas de controle, conforme 3.2.1. Ao atuar sobre estas estruturas altera-se, por conseguinte, todas as forças resultantes no modelo.

Como exemplo, considere um modelo de asa fixa, que para manter sua direção na rota armazenada o mesmo deva alterar seu curso atual em 20° , e para isso o controlador atua diretamente nos ailerons, e com isto o modelo altera seu ângulo de rolagem e as forças atuantes. Observa-se que as forças de sustentação inclinam na mesma direção da rolagem.

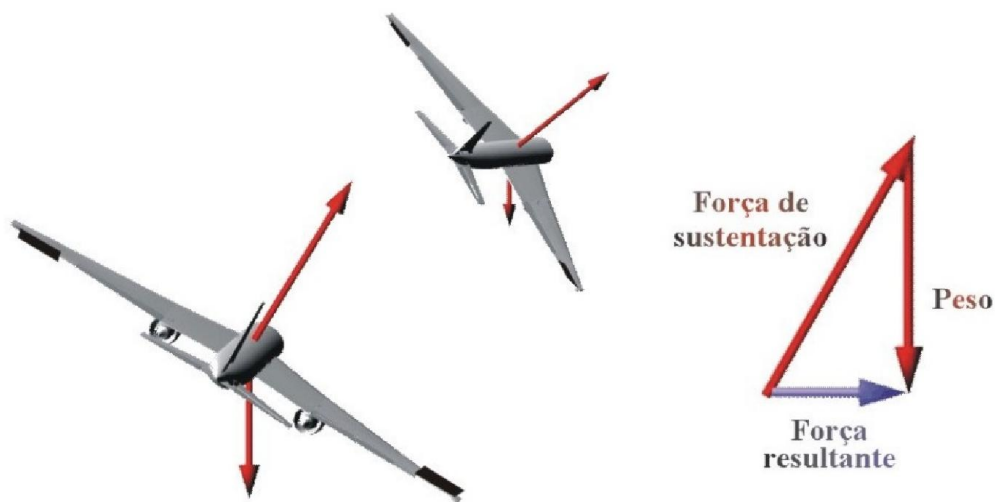


Figura 36 Curva glissada utilizando ailerons e forças resultantes (fonte (ARAÚJO., 2007)).

Decompondo-se as forças de sustentação e peso em vetores, pode-se observar que ao compensar o peso tem-se uma força resultante no sentido da rolagem. Esta força resultante é proporcional à rolagem, e altera o curso do modelo.

Considerando a equação aerodinâmica do perfil da asa, equação 3-1, em que a força de sustentação é resultado de suas variáveis, pode-se então alterar a sustentação de cada uma das asas, de acordo com a necessidade de controle do modelo atuando em qualquer uma de suas variáveis.

Para o modelo de asa fixa pode-se alterar o coeficiente de sustentação da asa ao movimentarmos as estruturas móveis da asa, alterando assim o perfil da mesma e, por conseguinte, seu coeficiente de sustentação. Assim, associa-se o resultado da equação 5-1 ao

coeficiente de sustentação da equação 3-1. Tem-se então a equação de sustentação do perfil associada ao controle conforme mostrado em 5-2.

$$L = \left(Kp \times e + Ki \times \int_0^t e \times dt + Kd \times \frac{de}{dt} \right) + c1 \times \left(\frac{1}{2} \times \rho \times V^2 \times A \right) \quad [5-2]$$

Desta maneira pode-se associar todos os controladores de atitude ao gerenciador de direção conforme a Figura 37. Ao se definir os parâmetros do controle pode-se controlar a atuação do ângulo máximo de rolagem ou arfagem.

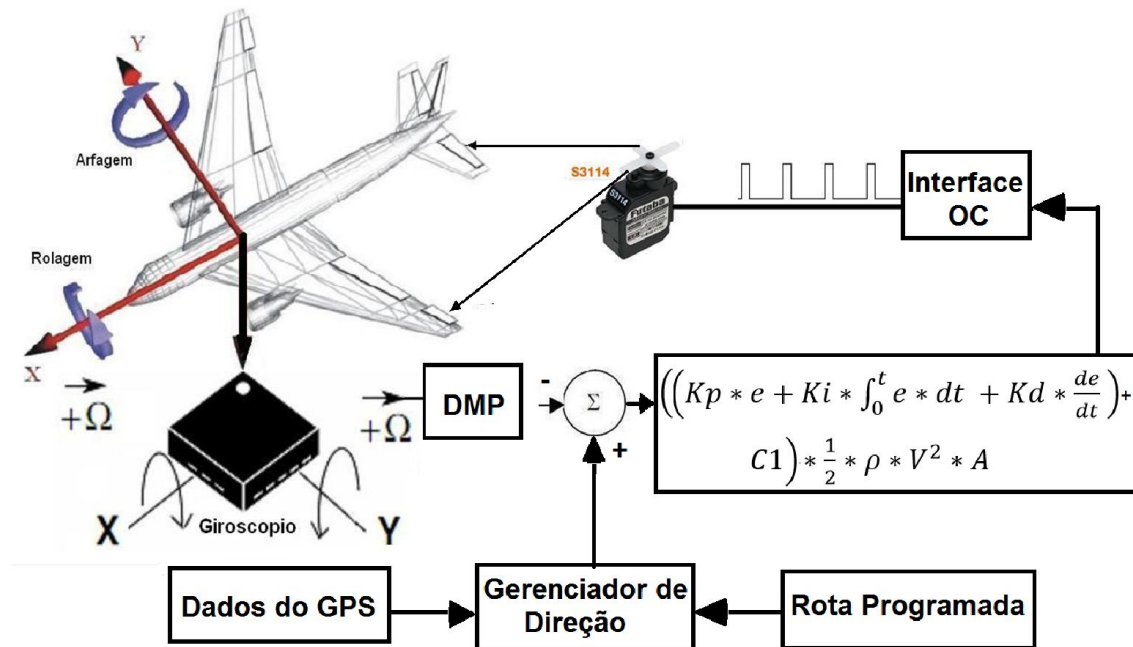


Figura 37 Controlador aplicado ao modelo de asas fixa.

5.7 Armazenamento de Dados

O sistema possui um *slot* para cartão microSD instalado em uma das portas SPI do DSP. Esta interface é configurada para trabalhar com velocidade de transmissão de 10 Mbps. O sistema implementa uma biblioteca fornecida pelo próprio fabricante do DSP para controle do cartão microSD com formatação FAT32.

A biblioteca é fornecida com código fonte para que possa ser configurada e alterada de acordo com a necessidade de implementação. Neste trabalho algumas características

relevantes são habilitadas como, FAT32, suporte a nomes extensos e manipulação simultânea de até 3 arquivos para permitir a expandir as aplicações com uso de armazenamento de dados.

Pelo fato de a interface ser serial e a formatação FAT32 demandar manipulação de um volume de dados relativamente grande em relação a cada *cluster* de armazenamento, o sistema implementa um espelho do arquivo em um *buffer* de 2 kbytes para reduzir o trânsito de dados, sendo que o *buffer* é descarregado no cartão assim que o mesmo for carregado a 50%. Com isto se reduz as intervenções com grandes volumes de dados o que demandaria mais tempo para escrita no cartão.

O sistema após o *reset* procura um arquivo de texto “CELG2.TXT”; caso o sistema não encontre, então o arquivo é criado no cartão; caso encontre o arquivo ele é aberto com permissão para escrita. Com o arquivo aberto o sistema passa a utilizar o buffer para gravar os valores das variáveis selecionadas em formato texto. Estes dados podem ser visualizados em qualquer editor de texto de qualquer sistema que ofereça suporte a FAT32.

Com o uso do cartão é possível transferir uma nova rota, carregar a configuração do modelo, ou armazenar uma quantidade de dados relativamente grande. Pode-se ressaltar que o arquivo aberto para gravação dos dados de voo é sempre o mesmo.

5.8 Sistema de Monitoramento

O *software* de monitoramento foi criado para atender às necessidades de controle já implementadas na placa, tais como protocolo de comunicação, armazenamento de imagens, criação, gerenciamento e acompanhamento das rotas, atualização de parâmetros de controle em tempo real, monitor e gravador de vídeo em tempo real.

O software de programação e acompanhamento de missão (SPAM), utiliza armazenamento em banco de dados para repetição do acompanhamento já gravado. Permite também o acompanhamento em tempo real da atividade de inspeção, através de imagens da janela de monitor que reproduz as imagens da câmera instalada no modelo diretamente na tela do computador; e acompanhamento de variáveis de navegação, tais como as coordenadas GPS, gráfico de altura de voo, quantidade de carga de bateria. Na Figura 38 observa-se a tela de configuração dos parâmetros do modelo.

Cada modelo é único e tem suas configurações de acordo com as características de voo, parâmetros de construção e alinhamento. Para atender estas necessidades o SPAM tem uma tela de configuração que pode armazenar todos os parâmetros de cada avião em uso.

Estas configurações podem ser transferidas ou conferidas no modelo a qualquer momento. Os parâmetros podem ser alterados com o modelo em voo, sendo que as alterações são aplicadas no instante que o dado é gravado no modelo. Os novos parâmetros podem ser inseridos de acordo com as necessidades do programa do modelo, a tela de configuração pode ser observada na Figura 38.

	Nome	Endereço	Tipo de dados	Valor Padrão	Valor do Dispositivo
▶	pitchGP	0	Int	500	400
	pitchGI	2	Int	30	30
	PitchGD	4	Int	100	100
	RollGP	6	Int	500	400
	rollGI	8	Int	50	50
	RollGD	10	Int	373	373
	LEMEGP	12	Int	50	50
	LEMEGI	14	Int	50	50
	LEMEGD	16	Int	50	50
	RPMGP	18	Int	100	100
	RPMGI	20	Int	50	50
	RPMGD	22	Int	50	50
	BATERIA_N_S	24	Char	3	3
	TIPO_BAT	25	Char	1	1
	TEMP_STATUS...	26	Int	2002	2002
	TEMP_STATUS...	28	Int	1000	1000
	TEMP_Instrumen...	30	Int	500	500

Figura 38 Tela de configuração de parâmetros do modelo

A Figura 39 ilustra uma aplicação do SPAM. Ela registra a imagem da câmera de vídeo e a rota seguida pelo modelo.

No capítulo seguinte serão detalhados os protocolos de comunicação e o padrão de dados recuperáveis do modelo, armazenados no cartão SD.



Figura 39: Imagem da câmera de bordo e acompanhamento de rota no SPAM

6 APLICAÇÃO AO MODELO DE ASA FIXA

Neste capítulo descreve-se o modo de implementação e aplicação da placa ao modelo de asa fixa, bem como o processo de testes em laboratório e os testes de campo.

6.1 Aspectos gerais

A implementação da placa a um aeromodelo real demanda ensaios de maneira a reduzir os riscos de danos materiais, sendo simuladores computacionais de aviões reais os mais indicados para a atual aplicação. O simulador utilizado neste trabalho foi o Flight Gear. O sistema também tem que passar por testes de acionamento, transmissão de rota, de atualização de configuração e validação dos modos de operação para que possa ser encaminhado ao modelo real.

6.2 Resultados de laboratório

O início dos testes é para confirmação de funcionamento das partes internas dos controladores de software da placa. O primeiro teste a se fazer é o de recepção, onde observa-se a recepção e decodificação do sinal PPM do receptor. A placa deve ser capaz de decodificar os sinais recebidos diretamente do controle remoto por meio de duas entradas PPM seriais para até 12 canais cada e/ou 6 entradas PWM para canais individuais.

A Figura 40 mostra a janela de comunicação entre o controle do aeromodelo e a placa de comunicação, sendo que a decodificação do canal é feita na placa extraindo do receptor o sinal PPM de acordo com o número de canais do transmissor. São mostrados 4 canais de comunicação (sinal enviado e sinal recebido).

A Figura 41 mostra o sinal enviado pelo controle remoto e recebido pela placa. O sinal em amarelo mostra o sinal modulado recebido do controle remoto e o sinal em verde mostra o sinal decodificado de um dos canais do receptor.

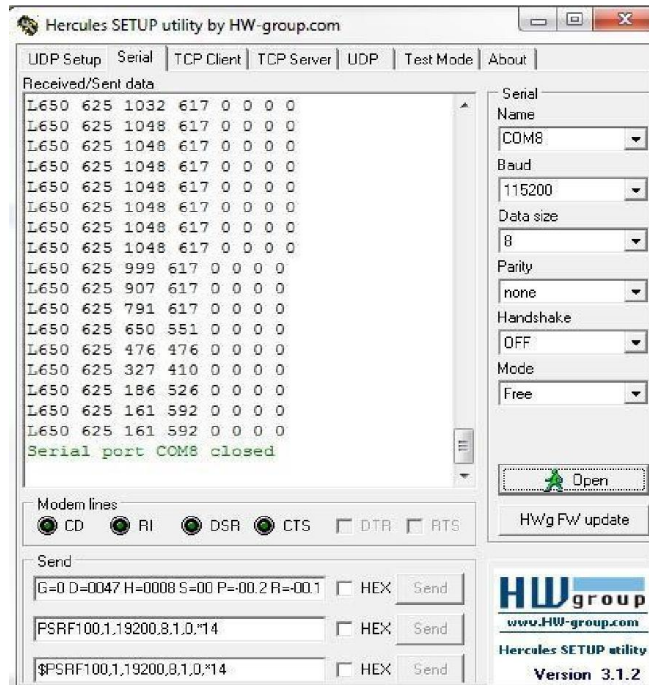


Figura 40 Janela de comunicação entre o controle do aeromodelo e a placa controladora

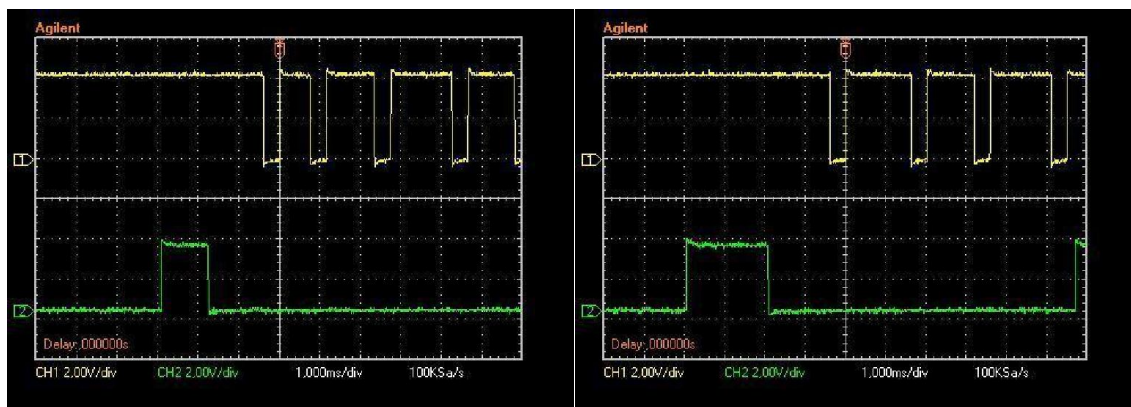


Figura 41 Sinal de controle (modulado e decodificado) recebido pela placa controladora

6.2.1 Teste de Acionamento de Servo-Motores

A placa tem 8 saídas PWM para controle direto dos servos e 1 saída PPM para um sistema de decodificação que inclui os 8 canais diretos na placa mais 4 que existem na lógica do programa, podendo assim o sistema controlar até 12 servos.

A Figura 42 mostra a montagem feita para a realização dos testes de acionamento dos servo-motores. São 4 (quatro) servos, responsáveis pelo acelerador, leme, profundor e ailerons.

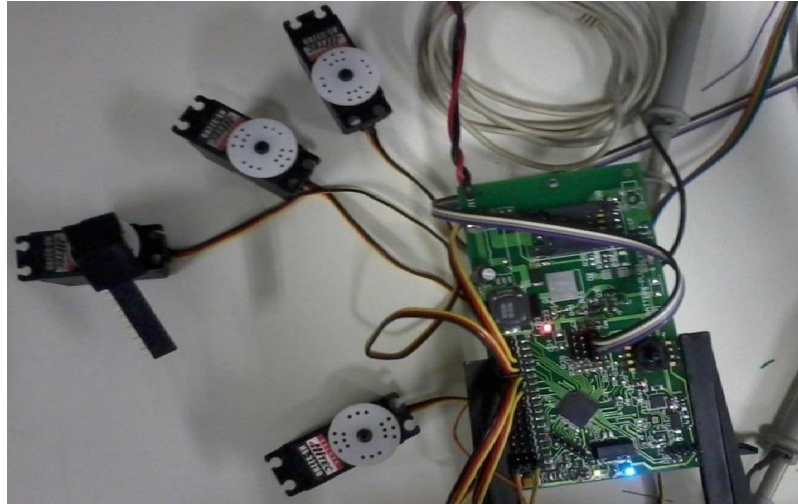


Figura 42 Acionamento de 4 servo-motores usando a placa controladora

A Figura 43 mostra a posição do manete de controle (*stick*), a posição correspondente do servo-motor e o sinal resultante no osciloscópio. O servo-motor está a 0 grau e o "*stick*" deslocado para cima. O osciloscópio mostra no canal 1 o sinal enviado pelo controle, com modulação por posição de pulso (modulação PPM) e o canal 2 mostra o sinal decodificado. A largura do pulso de controle do servo corresponde a 1,1 ms.

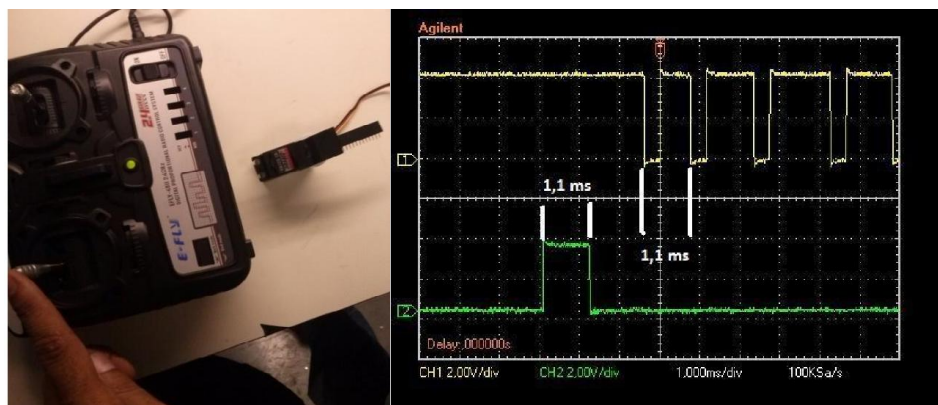


Figura 43 - Posição do *stick* e do servo (a 0 grau) e sinais modulado (modulação PPM) e decodificado

Na Figura 44 o servo está a 60° e o "stick" deslocado para baixo. No canal 1 do osciloscópio tem-se o sinal modulado enviado pelo controle e no canal 2, o sinal decodificado. A largura do pulso de controle do servo corresponde a 2 ms.

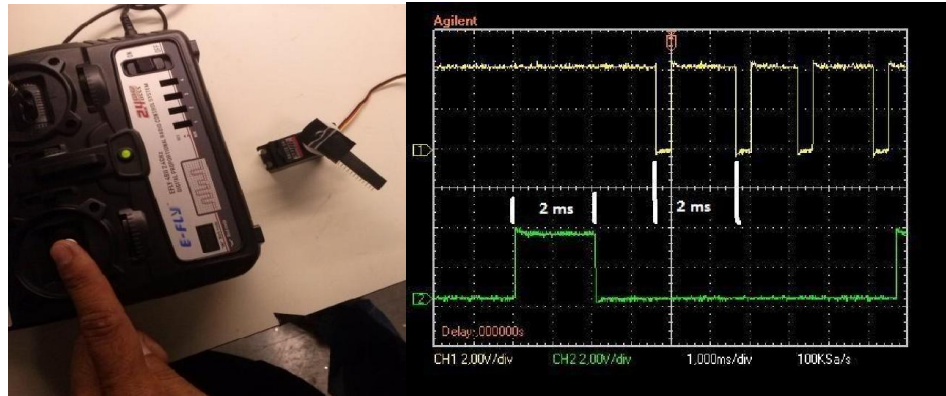


Figura 44-Posição do stick e do servo (a 60 graus) e sinais modulado (modulação PPM) e decodificado

Após verificar a recepção e atuação dos servos, observa-se o funcionamento do receptor de GPS e circuitos interligados ao mesmo. Para isso conectou-se o mesmo à porta serial correspondente, aguardou-se a sintonia de satélites, sendo que para obtermos dados de localização são necessários no mínimo 3 satélites para um referenciamento em 2D. Para o referenciamento em 3D são necessários um mínimo de 4 satélites. A quantidade de satélites interfere diretamente na precisão do posicionamento. Para verificação dos dados do GPS pode-se observar a continuidade dos dados oriundos do GPS, que são gravados no cartão, como latitude, longitude e altitude. Existem outros dados que são extraídos do GPS, como a quantidade de satélites que são utilizados, direção do movimento, entre outros, que por serem apenas para controle, não são gravados.

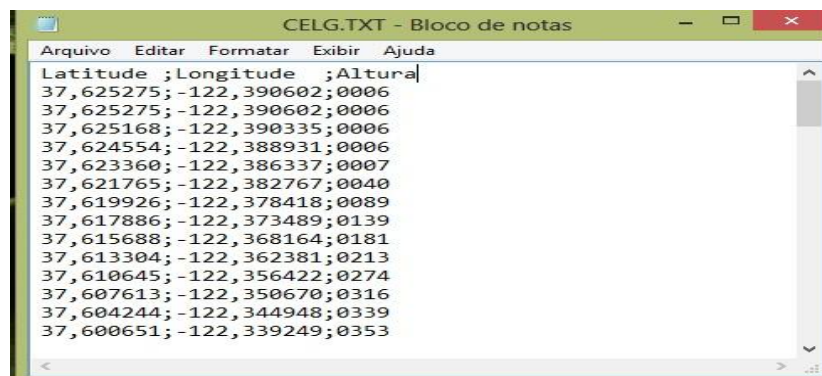


Figura 45 Arquivo CELG.TXT de armazenamento dos dados de voo

Para a verificação de funcionamento do cartão SD/MMC habilita-se a gravação dos dados no mesmo e posteriormente é feita uma leitura do cartão. Os dados mostrados na Figura 45 permitem concluir que o sistema está funcionando normalmente, uma vez que nem um erro foi detectado.

6.3 Simulador Flightgear.

Após confirmar o funcionamento das interfaces de controle do sistema, o próximo teste é o uso da placa para controlar um modelo no simulador digital. O uso de um simulador de voo permite a execução de vários testes na placa e no seu código de controle sem o risco de perdas ou acidentes. O simulador de voo escolhido é o *flightgear*.

Conforme o site do simulador (FLIGHTGEAR, 2015) “O objetivo do projeto *Flightgear* é criar uma infraestrutura sofisticada de simulação de voo para uso em ambientes acadêmicos e de pesquisa, para o desenvolvimento e busca de ideias interessantes no campo da simulação, e também uma aplicação para o usuário final”. Com este conceito, o simulador é totalmente flexível, podendo ser alterado para cada uma das necessidades de simulação, desde a aerodinâmica do perfil da asa à simulação de instrumentos de voo.

O flightgear é um simulador de voo gratuito, de código aberto e pode ser baixado diretamente da internet. Ele reproduz com muita fidelidade o voo de aviões como Cessna 172 e Piper J3 dentre muitos, é necessário um computador X86 com 2G de memória RAM, e sistema operacional Windows ou Linux podendo ser 32 ou 64 bits, sendo que deve ser baixada a versão correspondente ao sistema operacional utilizado.

Pelo motivo de já trabalhar com esse simulador desde os tempos da graduação, e pelo fato do mesmo já estar instalado em um computador virtual ainda em uso, optou-se por manter a instalação da versão 1.9.1 do simulador em uma máquina Windows XP de 32 bits para este trabalho. A versão atual do simulador já consta no site do desenvolvedor como 3.4, e pode ser utilizada seguindo o mesmo conceito de aplicação utilizado na versão 1.9.1.

Após a instalação do simulador pode-se observar que são poucos os modelos que constam na base de dados do simulador, considerando que naturalmente cada modelo tem suas características de voo, deve-se observar as características mais próximas do modelo real. O modelo escolhido para simulação é o RASCAL110, por ser um aeromodelo comercial e compatível com a aplicação do trabalho proposto. O modelo é selecionado na tela principal do simulador conforme Figura 46.

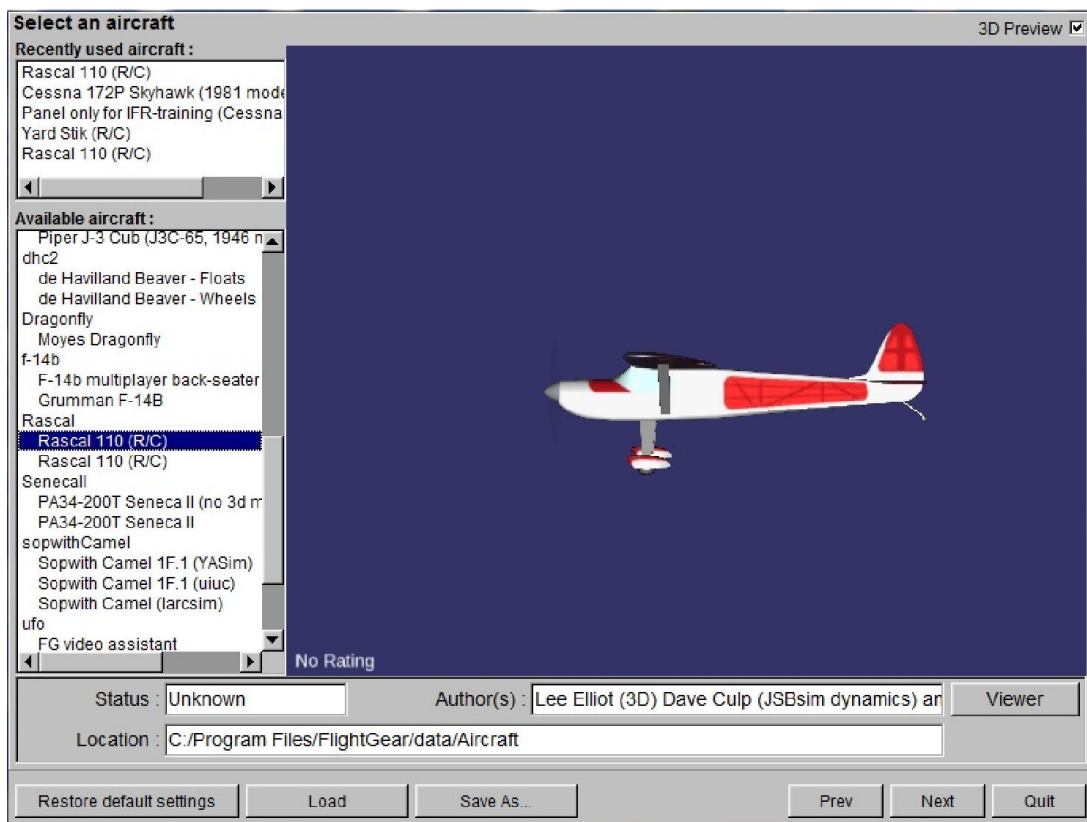


Figura 46 Tela de Seleção do modelo no simulador FlightGear.

Ressalta-se que este modelo deve ser instalado na base de dados do simulador, por meio de *download* fornecido na própria página do simulador e descompactando o arquivo diretamente na pasta Aircraft do simulador.

Foi instalado também um pacote de tradução do simulador para que os menus de configurações sejam apresentados em português, estes pacotes podem ser baixados diretamente da página do desenvolvedor.

Após selecionar o modelo deve-se configurar o local de voo da simulação ou cenário, como é definido para o simulador. São inseridos vários aeroportos na instalação como default do simulador, mas praticamente qualquer local do globo pode ser incluído para a simulação, bastando apenas baixar o pacote correspondente à localização da área a ser utilizada como cenário. Para fazer a instalação destes pacotes é necessário descompactar na pasta correta, conforme as instruções de instalação contidas no documento de instalação do pacote baixado.

Para a simulação neste trabalho utilizou-se um aeroporto instalado como default conforme pode-se observar na Figura 47. O aeroporto selecionado é o de São Francisco KSFO e a pista é selecionada como default.

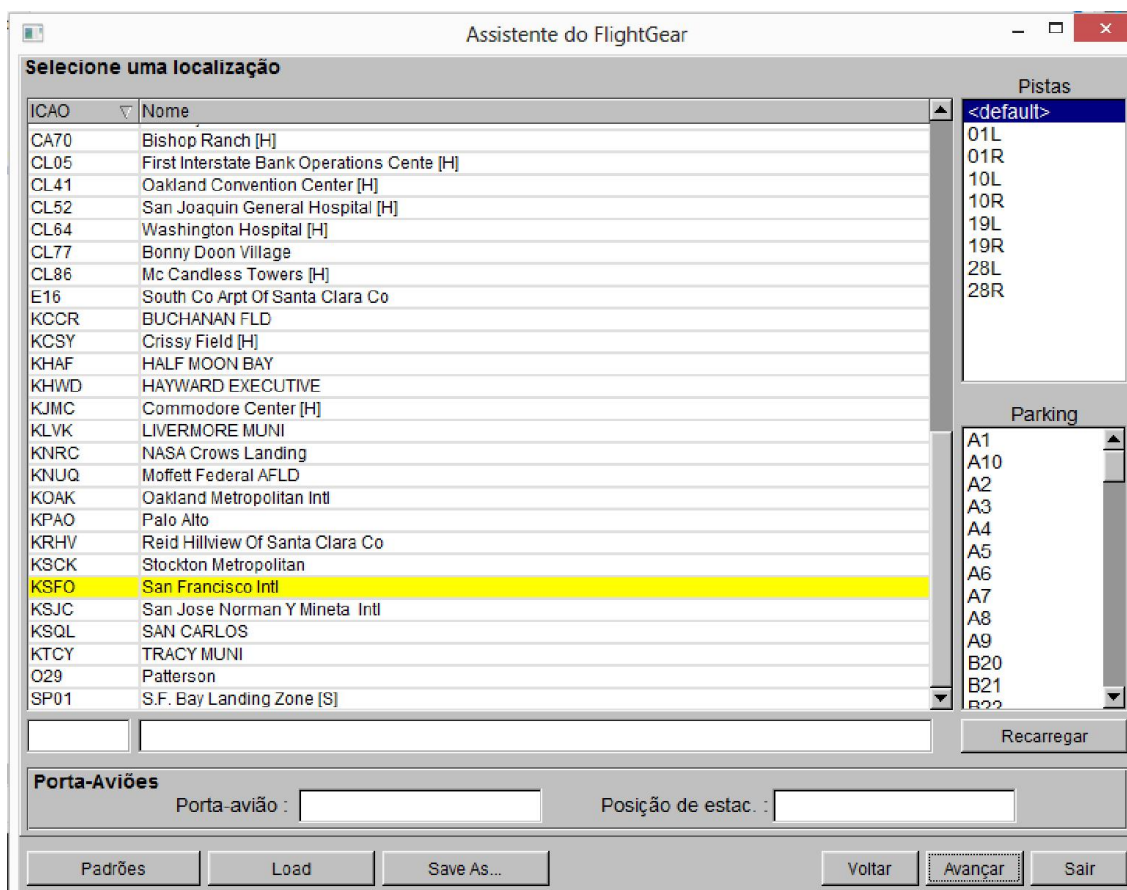


Figura 47 Configuração do aeroporto da simulação

Para simular o controle de voo de um avião utilizando a placa juntamente com o simulador, foi necessário definir o modo de simulação, pois os dados do GPS do avião simulado obedecem ao protocolo NMEA e, portanto, a placa já decodificaria naturalmente. Porém, os dados de estabilidade e inclinação do modelo devem ser tratados diretamente na placa e, como medida de contenção de portas de comunicação por parte da placa, a solução encontrada é a concatenação dos dados de GPS e atitude em uma única interface de saída.

Para tal, no simulador, é configurado uma interface de saída com os parâmetros de interfaceamento conforme a Figura 48, em que é definido um protocolo genérico para o simulador. Com isto pode-se criar um protocolo proprietário, com os dados que são necessários e com formatação adequada.

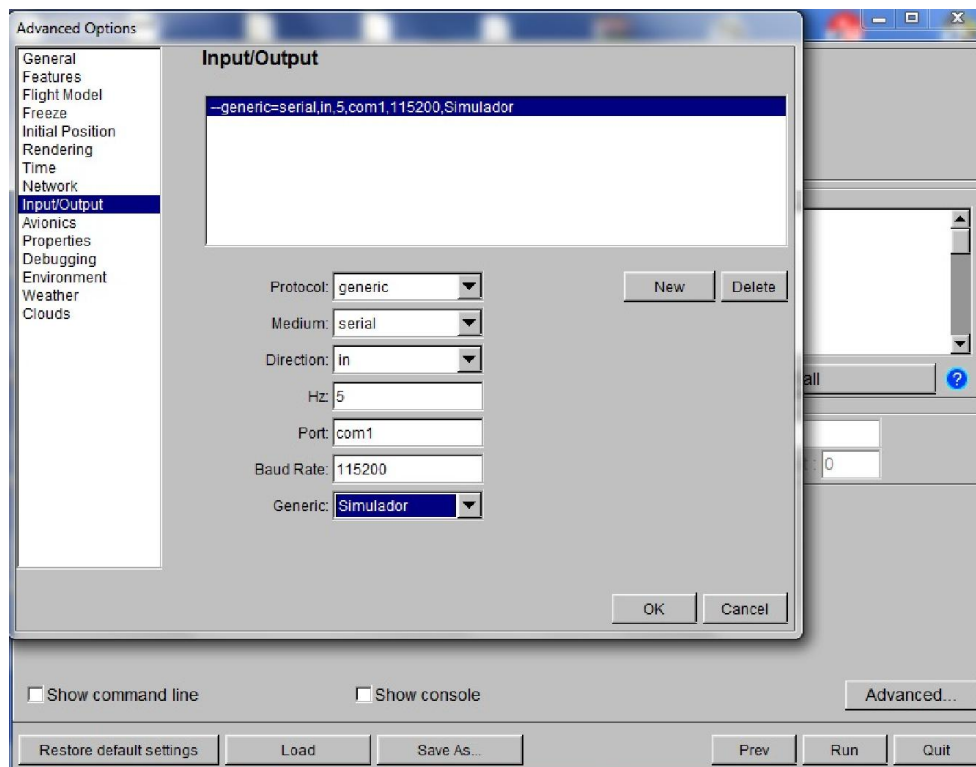


Figura 48 Configuração da interface de saída (FLIGHTGEAR, 2015).

Conforme pode ser observado na Figura 48, deve-se informar o tipo de mídia para a comunicação, em que se definiu como serial, tendo por referência que a placa controladora contém 4 portas seriais. O simulador é capaz de se comunicar por outros meios, como rede ethernet e armazenamento em arquivo.

Como nesta simulação pretende-se aproximar ao máximo da situação real de controle definiu-se a direção dos dados como saída. Poderia ser utilizada mais uma interface para o retorno dos dados de controle para o modelo simulado, porém com isto perderia-se a oportunidade testar a atuação das interfaces de saída para os servos e para a porta de saída PPM.

Define-se a latência dos dados na mesma latência utilizada no GPS com 5 Hz. Quanto maior este número, maior a precisão utilizada na simulação, por se tratar da frequência em hertz com que os dados são enviados.

Como já se definiu a mídia de saída como sendo portas seriais, deve-se selecionar na aba Port o número da porta utilizada para conectar o computador à placa controladora. A porta escolhida foi a COM1.

Na aba Baud Rate a velocidade em bps da porta serial utilizada é escolhida. A placa trabalha a 115200 bps, definindo assim o mesmo valor para o computador.

Na aba Generic é selecionado o nome do arquivo de protocolo, que define a comunicação; este arquivo é basicamente a definição dos dados utilizados conforme descrito a seguir.

As configurações dos parâmetros de simulação são feitas em arquivos discretos relativos a cada parâmetro associado ao modelo simulado. São arquivos com formato XML que podem ser alterados em editores específicos para linguagem XML. No arquivo Simulador.XML no anexo 9.5 são definidos os dados de GPS e atitude, bem como a formatação utilizada. Esta formatação deve ser obedecida tanto no simulador quanto na placa e, para que se possa alterar os dados ou formato dos mesmos, deve-se reescrever os códigos de decodificação na placa, para que os dados de saída do simulador possam ser decodificados na placa controladora. Este arquivo deve ser gravado na pasta Protocol para que o simulador possa acessá-lo.

Após conectar o simulador à placa considerando o sentido dos dados do simulador para a placa, deve-se construir o caminho inverso dos dados agora de controle da placa para o simulador, considerando que se deve manter a simulação a mais fiel à realidade possível. Pode-se extrair os sinais de controle dos servos por meio da porta de expansão PPM.

Para enviar o posicionamento dos servos ao simulador utiliza-se um adaptador que converte os sinais PPM para conexão direta à porta USB, este adaptador cria no sistema operacional do computador um joystick virtual. Este joystick virtual é utilizado no simulador para controlar o modelo em voo. O adaptador pode ser observado na Figura 49.



Figura 49 Cabo adaptador PPM/USB

Desta maneira o modelo simulado passa a ser controlado pela placa que recebe os dados de voo do simulador fechando assim uma malha de controle.

Este adaptador é conectado à porta de expansão PPM e ligado diretamente à USB do computador. A placa recebe também os sinais PPM do receptor que pode também controlar o voo no simulador selecionando o modo autônomo ou assistido. Os dados da placa são observados diretamente no SPAM, tal qual seria no modelo real, tornando assim a simulação uma representação fiel do controle em voo.

A Figura 50 apresenta um fluxograma dos dados da simulação, onde a placa real controla o aeromodelo do simulador.

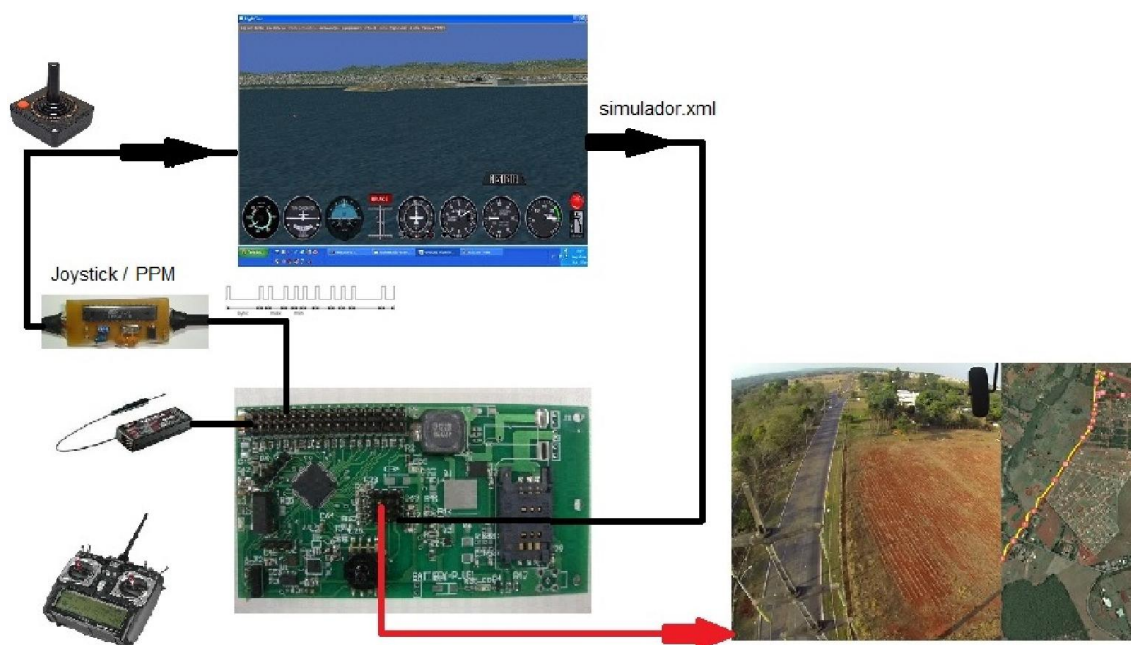


Figura 50 Fluxo de sinais para simulação da placa

Para iniciar a simulação, é gerada uma rota no SPAM, que é transferida para a placa pelo próprio SPAM. Após a transferência, são definidos os parâmetros iniciais dos controladores PID relativos aos ângulos de inclinação Pitch e Roll, bem como todos os dados de configuração, que podem ser alterados em voo. As alterações no comportamento podem ser observadas com facilidade pelas perturbações ao modelo em voo, como ventos e estabilidade do voo, todos os parâmetros de configuração do modelo podem ser reconfigurados de acordo com a necessidade de aplicação do modelo ou características de construção do mesmo.

O modelo é colocado em voo e nivelado manualmente; então o controle é transferido para a placa, e as alterações de comportamento do modelo são observadas para cada um dos parâmetros inseridos, observando sempre como principal fator a estabilidade da atitude do modelo em relação ao horizonte e ao acompanhamento da rota.

Para se avaliar a conclusão da etapa de simulação, são observados dois critérios mínimos de aceite, sendo eles:

- Parâmetro de estabilidade e resposta a perturbações aerodinâmicas a que o modelo será submetido em condições reais de voo, tais como vento e variação de pressão atmosférica.
- Interferência de ruídos eletromagnéticos e quantidade de tráfego de dados oriundos dos sistemas de comunicação, com observação aos tempos de resposta dos controladores.

Para o teste de estabilidade de atitude e resposta às perturbações de voo, a placa atendeu às necessidades de simulação após a sintonia dos valores de ganhos dos controladores PID de Pitch e Roll fazendo com que a aeronave se mantivesse em voo nivelado e direção constante, conforme pode-se observar na Figura 51.



Figura 51 Atuação da placa controladora em voo nivelado no simulador V1.9.1. (FLIGHTGEAR, 2015)

Para o critério de ruídos eletromagnéticos do sistema de transmissão e capacidade de tráfego de dados, pode-se concluir que a placa atendeu a todas as necessidades, não apresentando quaisquer alterações às configurações aplicadas.

Após a estabilização do modelo é feita verificação da atuação do controle de rota. Para essa verificação é gravada uma rota no modelo através do SPAM. O modelo é colocado em voo manual no simulador e transferido para o controle automático. Pode ser observado na Figura 52 que o modelo percorre a rota programada.

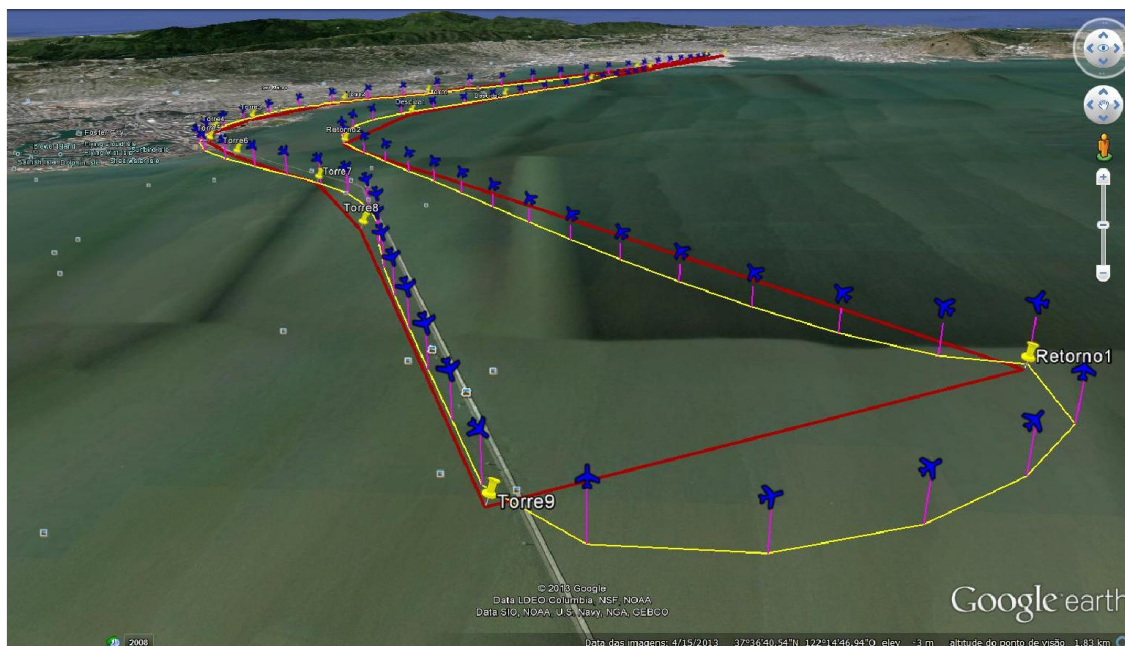


Figura 52 Rota programada e percorrida no simulador.

6.4 Resultados de campo

Uma fase não mais importante que qualquer outra, mas com um elevado grau de dificuldade, é a implementação da placa de controle no modelo real, uma vez que para cada modelo são levados em consideração diversas variáveis, tais como peso total de voo, diferença entre o peso de subida e o peso de descida, velocidade de estol (velocidade mínima para sustentação do modelo em relação à atmosfera), tempo mínimo de voo, razão de subida e descida, dentre tantas outras variáveis, para que, com parâmetros mínimos se pudesse sintonizar os coeficientes do controlador PID, convergindo o controle para a estabilidade.

Para minimizar a perda e/ou a avaria de equipamentos, definiu-se uma metodologia para a realização dos testes e alterações foram implementadas no programa de modo apenas a estabilizar o modelo em voo, sem um curso e altura definidos.

6.4.1 Coeficientes de *Offset*

Para instalar a placa no modelo real o mesmo deve ser montado em estrutura nivelada de maneira que o modelo corresponda ao voo nivelado, e a placa de controle deve ser instalada nivelada com o horizonte. Observando os dados de horizonte da placa, deve-se corrigir o posicionamento interno ao modelo, reduzindo assim o erro de offset. Este

procedimento deve ser repetido até que se encontre erros menores que $\pm 2^\circ$ de inclinação nos eixos de *pitch* e *roll*. Posterior aos parâmetros de solo, o modelo é posto em voo com controle manual. Imprime-se uma velocidade de voo 30% maior que a velocidade de estol e verifica-se o erro de *offset* em relação ao voo nivelado. Estes parâmetros devem ser considerados após extraídos de no mínimo três voos consecutivos em condições climáticas favoráveis. Levando-se em conta uma média das medidas, este procedimento deve ser repetido para toda alteração de posicionamento da placa interna ao modelo.

6.4.2 Coeficientes Controlador PI *Pitch*.

Os parâmetros de *offset* levantados são então inseridos na configuração da placa de modo que as leituras da Unidade de Médias Inerciais (IMU) sejam compensadas e apresentem apenas as inclinações do modelo tanto em solo quanto em voo.

Após inserir os parâmetros de *offset* nos controladores, então é efetuada a sintonia dos coeficientes de ganho do KP e KI do controlador Proporcional Integral (PI) de *pitch*. Os ganhos são inseridos com o modelo em voo e observando as respostas do controle para o voo nivelado. Após obter os coeficientes que estabilizam o modelo em voo nivelado, são inseridas perturbações em voo de maneira a certificar que os parâmetros são apropriados ao modelo. Para toda alteração de valores nestes coeficientes o modelo deve ser colocado em voo manual para testar as reações do controle em voo.

6.4.3 Coeficientes Controlador PI *Roll*.

Seguindo os mesmos procedimentos da regulação do controlador PI para *pitch*, os dados de configuração do controlador PI para *Roll* levantados são então inseridos na configuração da placa de modo que as leituras do IMU sejam compensadas e apresentem apenas as inclinações do modelo tanto em solo quanto em voo.

Após inserir os parâmetros de *offset* nos controladores, então é efetuada a sintonia dos coeficientes de ganho do KP e KI do controlador de *roll*, de forma similar aos testes com os ganhos são inseridos com o modelo em voo na tela de configuração do SPAM, observando as respostas do controle para o voo nivelado. Após obter os coeficientes que estabilizam o modelo em voo nivelado, são inseridas perturbações em voo de maneira a certificar que os

parâmetros são apropriados ao modelo. Para toda correção nestes valores, o modelo deve ser colocado em voo manual para testar as reações do controlador em voo. Este procedimento de teste deve ser reiniciado a partir dos dados de *offset* para toda vez que a configuração for alterada tanto para *pitch* quanto para *roll* para assegurar que o controlador será capaz de corrigir a atitude do aeromodelo em tempo hábil.

A Figura 53 apresenta as diversas curvas em que o modelo voou intercalando entre voo assistido e autônomo. Na parte superior observa-se um trecho em linha reta em que o modelo percorre sem a intervenção do operador, demonstrando a estabilidade do sistema.



Figura 53 Teste de estabilização.

Após a sintonia dos controladores com o modelo ainda em voo controlado e com altitude elevada, o modelo é colocado em grandes inclinações tanto para *pitch* e *roll* de modo a se observar o tempo de recuperação para o voo nivelado. Estes testes são efetuados para se compreender os limites de controle e de navegabilidade do modelo.

6.4.4 Teste de rota

Com os controladores PI já sintonizados para o modelo, então o programa é regravado com as funções do gerenciador de rota habilitado, é transferida uma rota para o modelo por meio do SPAM, e o acompanhamento é habilitado direto no SPAM com a gravação do trajeto efetivado para acompanhamento e avaliação posterior.

Como rota, definiu-se as cabeceiras da pista como trajeto escolhido para melhor observação visual do voo, e possível intervenção caso necessário. Na Figura 54 pode-se observar que o modelo percorreu a rota de maneira consistente por mais de uma vez habilitando assim o envio do modelo em uma rota efetiva, como teste final da etapa de teste de estabilidade do controle já instalado no modelo.



Figura 54 Rota com dois pontos.

Após os testes preliminares de estabilização aplicou-se uma rota programada pelo SPAM para simular uma rota de forma a ser monitorada como deve ser o uso do modelo de asa fixa. Na Figura 55 observa-se em vermelho, a rota programada e em amarelo a rota efetuada, o que demonstra diante dos resultados o bom funcionamento restando poucos ajustes a serem feitos nos controladores PI de *Pitch* e *Roll* para reduzir a ação brusca na correção da direção do modelo.

Os testes demonstraram que o modelo seguiu a rota de maneira satisfatória. A precisão do voo, porém, para a aplicação proposta, indica a necessidade de um melhor direcionamento do modelo em relação à direção do objetivo.

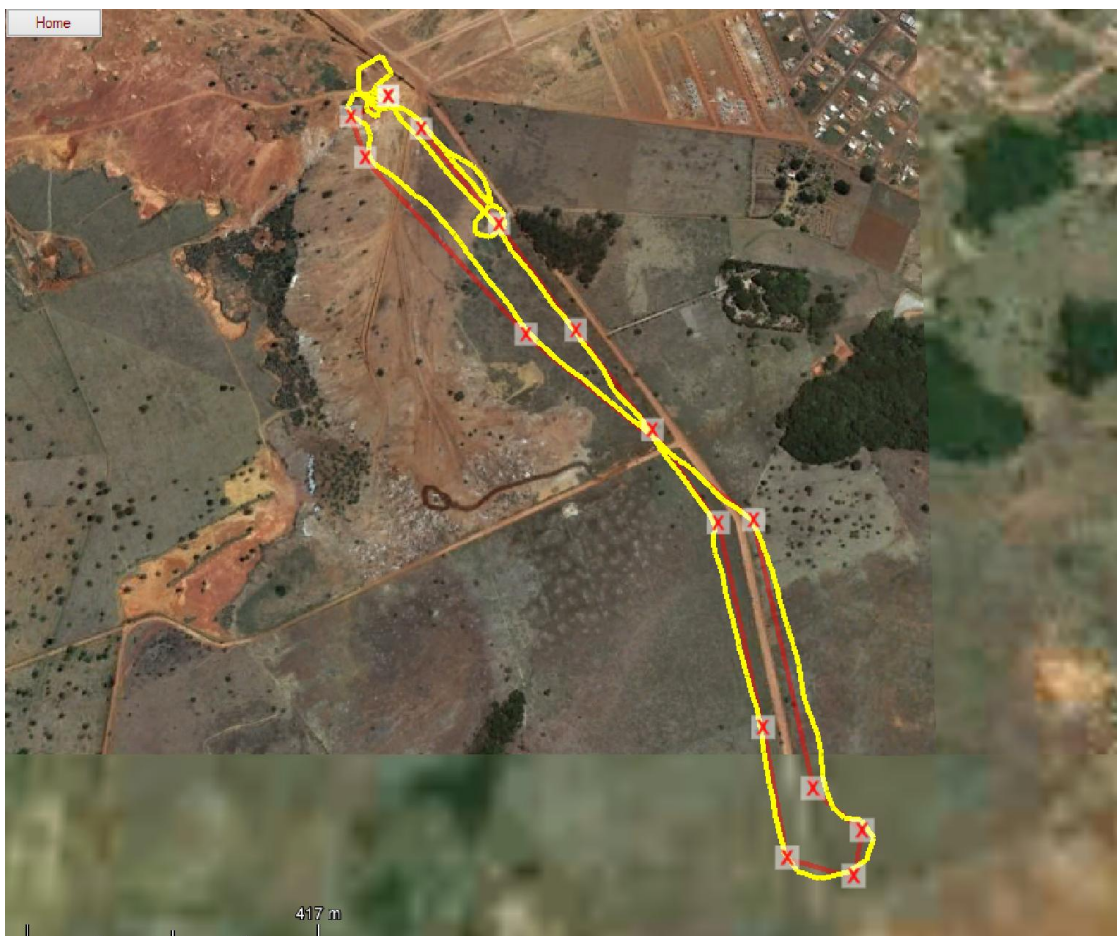


Figura 55 rota percorrida em modo autônomo.

Uma análise do código desenvolvido mostrou que o modelo calculava a direção com base na posição atual em relação ao próximo ponto de passagem, (a Figura 56 exemplifica como o cálculo é efetuado linhas em azul). Com este modo de recálculo, o modelo pode se afastar muito da linha central da rota (em vermelho), isso devido ao modelo estar com um ângulo pequeno fora da linha central no início do seguimento, e esse erro se torna acumulativo no decorrer do trajeto.



Figura 56 Pontos de recálculo da rota

Para se reduzir o erro acumulado devido ao recálculo com base no próximo ponto de passagem, é feita a criação em tempo real de subpontos na linha central para o posterior recálculo da direção. Com esta correção espera-se reduzir a distância do modelo da linha central da rota devido ao fato de que com o distanciamento da rota o erro aumenta substancialmente e a correção tem uma maior ação, diferente no modelo em uso que a correção é sempre em relação ao ponto de passagem.

Para o cálculo dos subpontos, é necessário que se insira os dados de latitude e longitude do próximo e do último ponto de passagem. Estes dados são computados na função distância inclusa no anexo 9.4. Nesta função é extraída a diferença entre os dois pontos da rota; esta diferença representa a variação numérica entre os pontos da rota considerando individualmente cada variável latitude (LatRT) e longitude (LonRT). O próximo passo é calcular a distância do modelo ao próximo ponto de passagem. Do valor da distância é subtraído o valor de aproximação, que pode ser configurável via SPAM. Este valor serve de referência para calcular um subponto com a mesma distância em metros, à frente do modelo. A Figura 57 exemplifica como seria o recálculo dos pontos intermediários e a correção da rota.



Figura 57 Subpontos da rota para recálculo.

Encontrada a distância do subponto em relação ao próximo ponto de passagem, a mesma é aplicada às variáveis LonRT e LatRT. Assim o valor é adicionado ao último ponto de passagem dando origem assim ao ponto de passagem virtual. Com o ponto de passagem virtual e a localização do modelo pode-se então calcular o erro de direção em relação ao ponto virtual calculado. O ponto virtual sempre estará localizado a uma distância que pode ser configurada a partir de 15 metros à frente do modelo. A função orientador é responsável pelo cálculo desta direção e seu código está no Anexo 9.4. Na Figura 58 pode-se observar o trajeto percorrido pelo modelo na linha em amarelo. Após o modelo ser desviado por uma corrente de ar (círculo em azul) o modelo retorna para a rota programada (linha em vermelho).

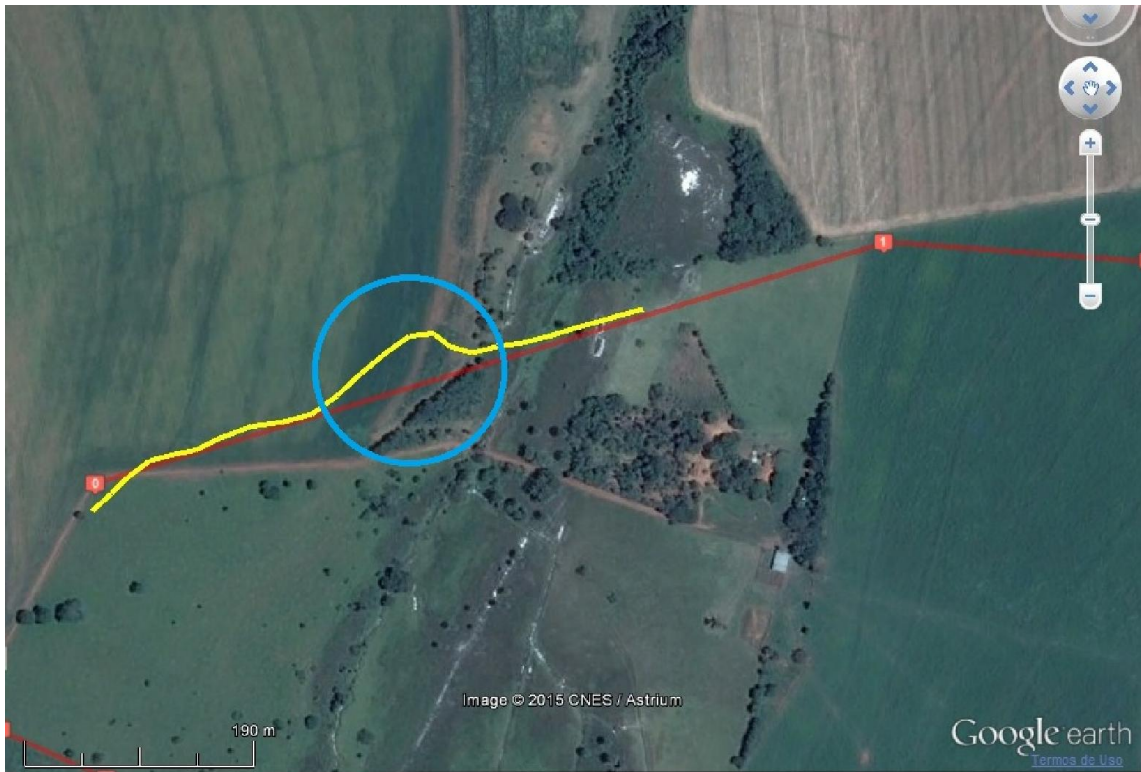


Figura 58 Palca com uso do recálculo de rota com subpontos.

7 CONCLUSÃO

Pode-se concluir que a placa atendeu ao objetivo de estabilizar e conduzir o modelo por uma rota pré-programada, e que o software dedicado SPAM é capaz de criar, alterar e reprogramar a rota no modelo seguindo os parâmetros de uso já mencionados.

O SPAM pode reprogramar o software da placa, para atualizar ou acrescentar novas funções dispensando o envio do modelo ao laboratório.

Dos objetivos propostos, sendo eles o projeto da placa e sua implementação, observou-se que o projeto da placa fora dividido em desenvolvimento da placa e escolha de seus componentes, e software da placa com sua integração com software de controle. A implementação foi dividida em duas aplicações distintas sendo elas, aplicação da placa em modelo de asa fixa e modelo de asa móvel.

No projeto da placa o desafio maior foi a compatibilidade magnética, sendo a maior influência das linhas de transmissão no sistema de guiamento por se referenciar no campo magnético da Terra, o que impediria a aplicação da placa ao uso pretendido. Ressalta-se também a imunidade à interferência eletromagnética induzida nas trilhas com uso de malha de proteção e capacitores de desacoplamento.

No decorrer do desenvolvimento observou-se a possibilidade de substituição dos sensores inicialmente escolhidos, por sensores com melhores respostas e maior quantidade e qualidade de literatura disponível para sua compreensão e aplicação, com maior disponibilidade no mercado, na criação da segunda versão de placa foi preterido que os sensores estivessem em modulo separado possibilitando assim a atualização dos sensores, seguindo os mesmos critérios de roteamento da primeira placa para imunidade eletromagnética.

Em relação ao software da placa destaca-se a não conclusão do mesmo em sua totalidade.

Foram concluídas: a integração com software de controle, a integração com os sensores da placa e interfaces, a implementação do sistema de *bootloader*. No entanto ainda não foi feita a integração com o módulo GPRS, que é responsável pela redundância na comunicação não afetando na aplicação da placa.

A implementação da placa no modelo de asa fixa foi concluída em sua totalidade. Ressalta-se que para novas versões do software devem ser avaliadas as aplicações de novas

funções inerentes à placa que possam afetar a aeronavegabilidade do modelo bem como a estabilidade do sistema como um todo.

A implementação da placa ao sistema de asa móvel não foi totalmente concluída. A avaliação dos dados de controle apontou que a respostas às perturbações de estabilidade não permitiram o voo do modelo devendo ser revistas variáveis referentes aos ganhos dos controladores PID, tempo de leitura e processamento dos sensores.

Para aprimoramentos futuros ou novos trabalhos sugere-se a possibilidade de implementação da placa a um sistema de asa móvel e a conclusão do software de controle com a implementação do sistema de GPRS, o que permitirá o acompanhamento do sistema na internet.

8 REFERÊNCIAS

- AMARAL, A. B. (Agosto de 2013). Desenvolvimento e uso de veículos aéreos não tripulados na inspeção de sistemas elétricos de potência. *P&D - Revista Pesquisa e Desenvolvimento*, pp. 86-88.
- ANEEL. (2011). Avanços Tecnológicos no Setor Elétrico. *Revista Pesquisa e Desenvolvimento da ANEEL*.
- ARAUJO., M. R. (18 de Junho de 2007). Projeto Simulação e Controle de Veiculo Aereo Nao tripulado Tail-Sitter. *Dissertação de Mestrado em Sistemas Mecatrônicos*. Brasília.
- BORGES. (20 de Setembro de 2015). *borgescorporation.blogspot.com.br*. Acesso em Agosto de 2015, disponível em Borges Corporation: <http://borgescorporation.blogspot.com.br/2013/05/filtro-de-media-movel.html>
- CALACHE, D. C. (11 de Junho de 2013). Caracterização de um Acelerômetro Baseado em Sistemas Microeletromecânicos (MEMS). *Projeto de graduação*. Rio de Janeiro, Rio de Janeiro.
- ESCOSTEGUY, J. P. (30 de Novembro de 2009). Projeto Vant. 4º *Workshop Nanoaeroespacial*. São José dos Campos, Sao Paulo.
- FLIGHTGEAR. (06 de 08 de 2015). *Flightgear*. Acesso em Agosto de 2015, disponível em Flightgear: <http://www.br.flightgear.org/sobre-flightgear/>
- FRANCISCO, P. S. (12 de 03 de 2012). *www.portalsaofrancisco.com.br*. Acesso em 12 de 03 de 2012, disponível em Portal São Francisco.: <http://www.portalsaofrancisco.com.br/alfa/como-funciona-o-aviao/como-funciona-o-aviao-1.php>
- INVENCENCE. (19 de 08 de 2013). MPU6050. *Datasheet MPU-6000 and MPU-6050*, pp. 1-52.
- JORGE, C., ANDRE, L., & INAMASU, R. (30 de Maio de 2011). Desenvolvimento de um VANT totalmente configurado para aplicações em Agricultura. *Simpósio Brasileiro de Sensoriamento Remoto*.
- MICROCHIP. (2011). dsPIC33EP512/24E Bootloader. *datasheet*, pp. 1-8.
- MICROCHIP, T. (2011). 16-bit MCU and DSC Programmer's Reference Manual. EUA.
- OLIVEIRA, F. A. (2010). Seminario INternacional de VANTS. *DCA - BR*.

- OLLIW. (17 de Setembro de 2013). *IMU Data Fusing: Complementary, Kalman, and Mahony Filter*. Acesso em 10 de 07 de 2015, disponível em Olliw: <http://www.olliw.eu/2013/imu-data-fusing/>
- ONS, O. N. (2011). *www.ons.org.br*. Acesso em 13 de Maio de 2015, disponível em ONS: http://www.ons.org.br/download/biblioteca_virtual/publicacoes/dados_relevantes_2011/arquivos/Cap7/7graficos/7e-Extensao-de-linha-de-transmissao-do-SIN-km-grafico.jpg
- PICCIONI, C. A., & Oliveira, R. S. (Junho de 2002). Introdução ao DSP. *Introdução ao DSP*. Florianópolis, Junho.
- SOARES DE LIMA, S. R. (2005). INTEGRAÇÃO GPS/INS UTILIZANDO SENSORES INERCIAIS BASEADOS EM SISTEMAS MICROELETROMECÂNICOS (MEMS). *Tese (Tese de Doutorado) Universidade Federal do Parana*. Curitiba , Parana.
- WEBER, L. (2014). SISTEMA SEM FIO PARA MEDIÇÃO EM TEMPO REAL DE VARIÁVEIS CINEMÁTICAS. *Trabalho de Conclusão de Curso (Graduação) Universidade Federal do Parana*, 1-66. Curitiba, Parana.

9 ANEXO

9.1 Sentença NMEA183 GPGBA

M = units of geoidal separation, meters

x.x = Age of Differential GPS data (seconds)

xxxx = Differential reference station ID

eg3. \$GPGBA,hhmmss.ss,lll.ll,a,yyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh

- 1 = UTC of Position
- 2 = Latitude
- 3 = N or S
- 4 = Longitude
- 5 = E or W
- 6 = GPS quality indicator (0=invalid; 1=GPS fix; 2=Diff. GPS fix)
- 7 = Number of satellites in use [not those in view]
- 8 = Horizontal dilution of position
- 9 = Antenna altitude above/below mean sea level (geoid)
- 10 = Meters (Antenna height unit)
- 11 = Geoidal separation (Diff. between WGS-84 earth ellipsoid and mean sea level. -=geoid is below WGS-84 ellipsoid)
- 12 = Meters (Units of geoidal separation)
- 13 = Age in seconds since last update from diff. reference station
- 14 = Diff. reference station ID#
- 15 = Checksum

eg3. \$GPRMC,220516,A,5133.82,N,00042.24,W,173.8,231.8,130694,004.2,W*70

- 1 220516 Time Stamp
- 2 A validity - A-ok, V-invalid
- 3 5133.82 current Latitude
- 4 N North/South
- 5 00042.24 current Longitude
- 6 W East/West
- 7 173.8 Speed in knots
- 8 231.8 True course
- 9 130694 Date Stamp
- 10 004.2 Variation
- 11 W East/West
- 12 *70 checksum

Fonte: <http://aprs.gids.nl/nmea/>

9.2 Função de Interrupção Timer1

```
void __attribute__((interrupt,no_auto_psv)) _T1Interrupt( void )//tempo de int. =1ms
{
    UNSIGNED CHAR C,SOMA,CHKSUM,D ;

    TMILLIS++;

    IF(--TPCT==0){
        WHILE (U2STABITS.URXDA)
        {
            GPSPCT[IPAC++] = U2RXREG;
        }
        STATUS_CON = STATUS_CON|GPS_SERIAL;
        GPSPCT[IPAC]=0;
        IPAC=0;
        LED=0;//*****
    }
    IF(DELAYEEP){DELAYEEP--;}
    IF(TIME_PTGPS){ TIME_PTGPS--; }
    IF(TEMPO_ATUALIZA_ATT){TEMPO_ATUALIZA_ATT--;}
    IF(TEMPO_ATUALIZADOR_ROTA){TEMPO_ATUALIZADOR_ROTA--; }
    IF(TEMPO_ATUALIZADOR_STATUS){TEMPO_ATUALIZADOR_STATUS--; }
    IF(TMMC)TMMC--;          // USADO NA ATUALIZACAO DO CARTAO
MMC
    //IF(T9150)T9150--;          // USANO NO CONTROLE DE VERIFICACAO
DO9150
    IF(TPID){TPID--;}
    ELSE{TPID=TEMPO_DT_PID; STATUS_CON=(STATUS_CON | DT_PID); }

    IF(TEMPO1++>10)
    {
        //LED =~ LED; TEMPODENTAL.COM.BR
        TEMPO1=0; }

    //LED !=LED;
    IFS0BITS.T1IF = 0;
}
```

9.3 Coordenador de direção

```
void distancia (double flatO, double flonO, double flatD, double flonD)
{
double x,y,calc,d;
x = 68.8 * (flatD - flatO);
y = 68.8 * (flonD - flonO) * cos(flatO/57.3);
d=sqrt((x*x) + (y*y))*1609.344;
Distancia=(unsigned long)d;
calc=atan2(y,x);
calc= calc*180/3.1415;
if(calc<=1){
    calc=360+calc;
}
Direcao=calc;
}
```

9.4 Criação de subpontos

```
void distancia(double flatO, double flonO, double flatD, double flonD)
{
double x,y,calc,d;

x = 58.8 * LatRT; //68.8
y = 58.8 * LonRT * cos(flatO/57.3);
x = 58.8 * (flatD - flatO); //68.8
y = 58.8 * (flonD - flonO) * cos(flatO/57.3);
d=sqrt((x*x) + (y*y))*1852;//1609,344;
LatRT=(flatD - flatO)/d;
LonRT=(flonD - flonO)/d;
Distancia=(unsigned long)d;
Percurso=Distancia;
calc=atan2(y,x);
calc= calc*57.295;
if(calc<=1){
    calc=360+calc;
}
Direcao=calc;
}
```

/******

* esta funcao calcula a distancia e a direcao entre duas cordenadas considerando
pontos de destino intermediaries

*****/

```
void orientador(double flatO, double flonO, double flatD, double flonD)
{
```



```

double x,y,calc,d;

x = 58.8 * (flatD - flatO); //68.8
y = 58.8 * (flonD - flonO) * cos(flatO/57.3);
d=sqrt((x*x) + (y*y))*1852;//1609,344;
Distancia=(unsigned long)d;
if(Distancia > Config.Aproximacao)
{
    flatD=flatD-(LatRT*(Distancia -Config.Aproximacao));
    flonD=flonD-(LonRT*(Distancia -Config.Aproximacao));
    x = 58.8 * (flatD - flatO); //68.8
    y = 58.8 * (flonD - flonO) * cos(flatO/57.3);
}
calc=atan2(y,x);
calc= calc*57.295;
if(calc<=1){
    calc=360+calc;
}
Direcao=calc;
}

```

9.5 Arquivo de Configuração do ‘Simulador.XML’

```

<?xml version="1.0"?>

<PropertyList>

<generic>

<output>
<line_separator>newline</line_separator>
<var_separator>newline</var_separator>

<chunk>
<name>pos-latitude-deg</name>
<type>float</type>
<format>A%03.6f</format>
<node>/position/latitude-deg</node>
</chunk>

```

```

<chunk>
  <name>pos-longitude-deg</name>
  <type>float</type>
  <format>O%03.6f</format>
  <node>/position/longitude-deg</node>
</chunk>

<chunk>
  <name>speed</name>
  <format>G%d</format>
  <node>/velocities/airspeed-kt</node>
</chunk>

<chunk>
  <name>heading</name>
  <type>float</type>
  <format>D%03.1f</format>
  <node>/orientation/heading-deg</node>
</chunk>

<chunk>
  <name>pos-alt</name>
  <format>H%03d</format>
  <node>/position/altitude-ft</node>
</chunk>

<chunk>
  <name>pitch angle</name>
  <format>P%03.1f</format>
  <type>float</type>
  <node>/orientation/pitch-deg</node>
</chunk>

<chunk>
  <name>bank angle</name>
  <format>R%03.1f</format>
  <type>float</type>
  <node>/orientation/roll-deg</node>
</chunk>

<chunk>
  <name>engine</name>
  <format>E%02d X</format>
  <node>/engines/engine[0]/rpm</node>
</chunk>
</output>
</generic>
</PropertyList>

```

9.6 Placa versão 1.2

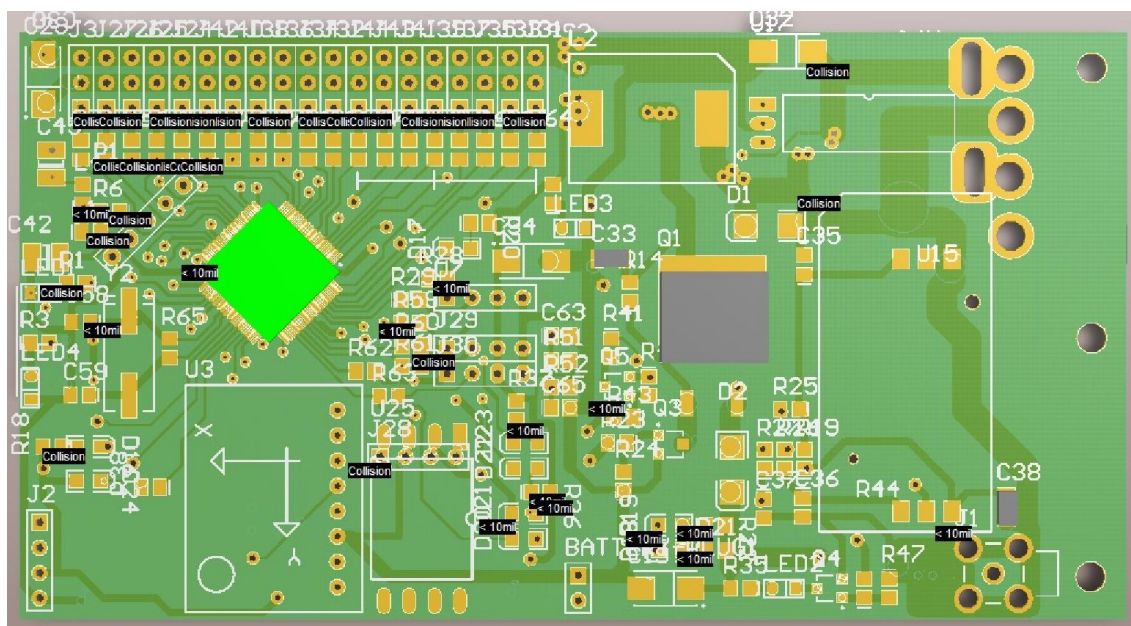


Figura 59 Vista 3D da placa versão 1.2

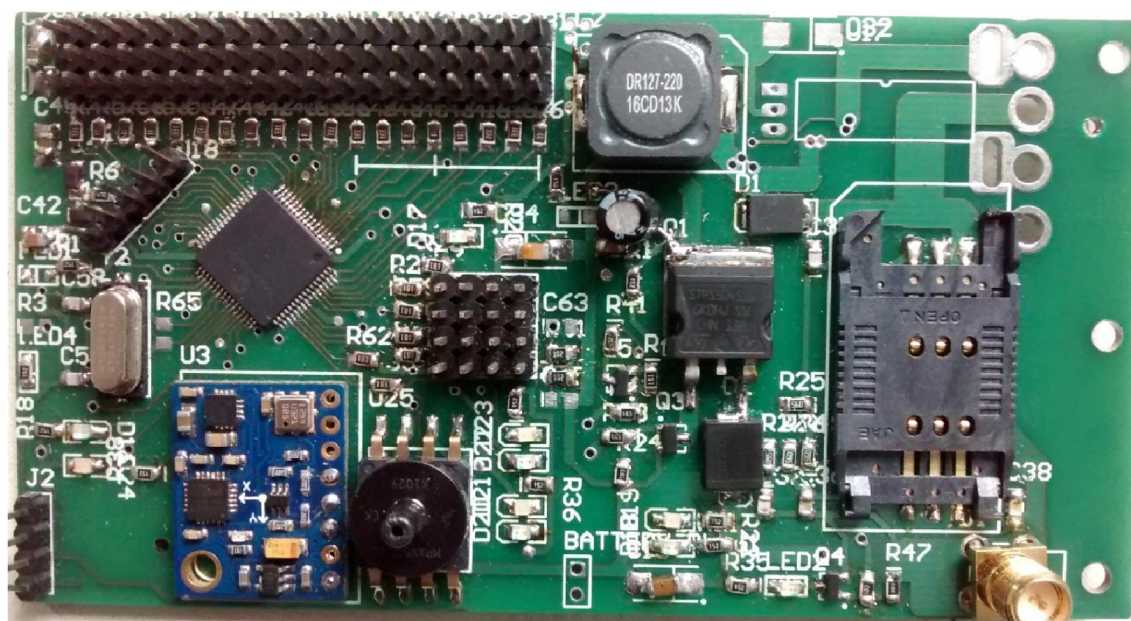


Figura 60 Placa montada versão 1.2 *layout* superior



Figura 61 Placa montada versão 1.2 *layout* inferior

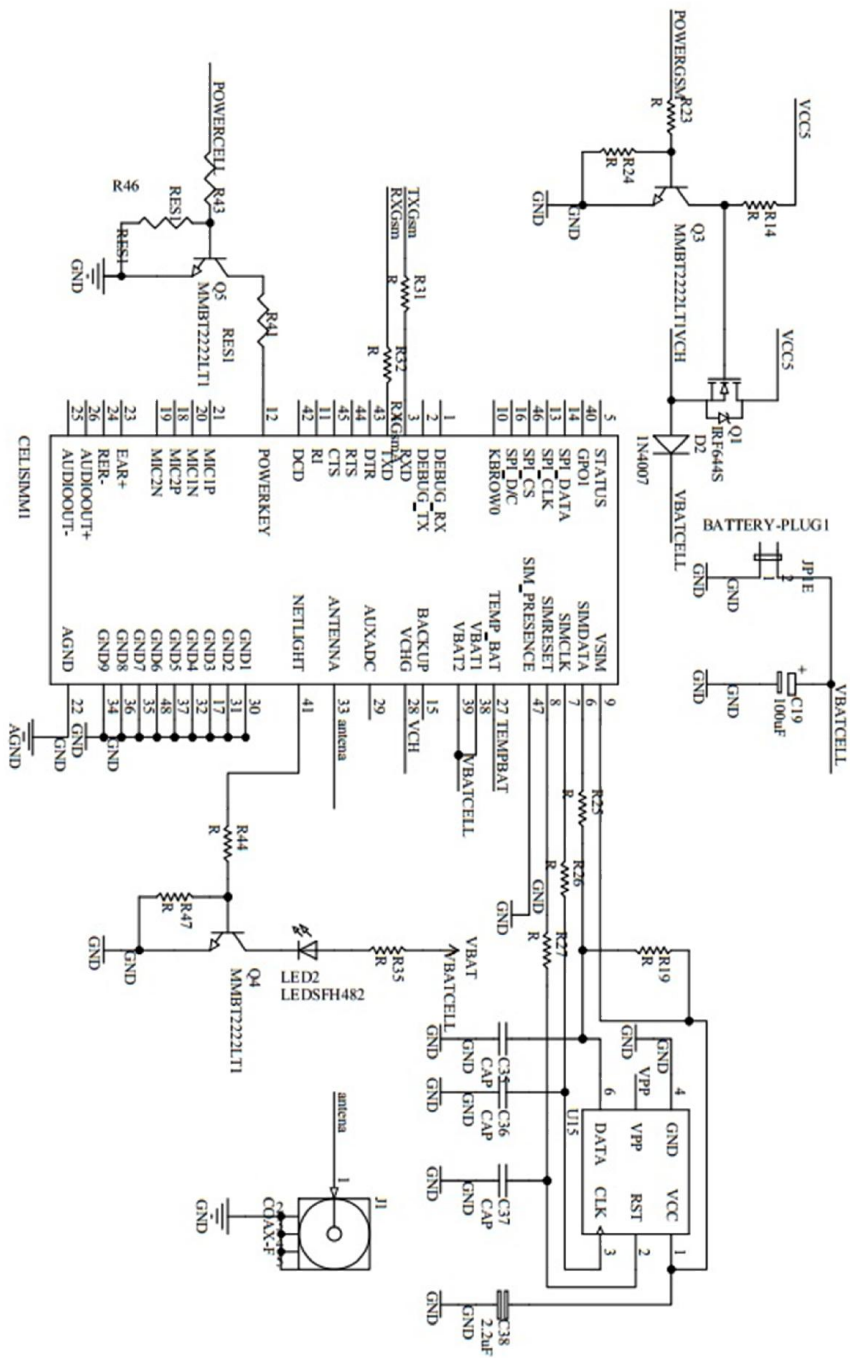


Figura 62 Módulo GPRS

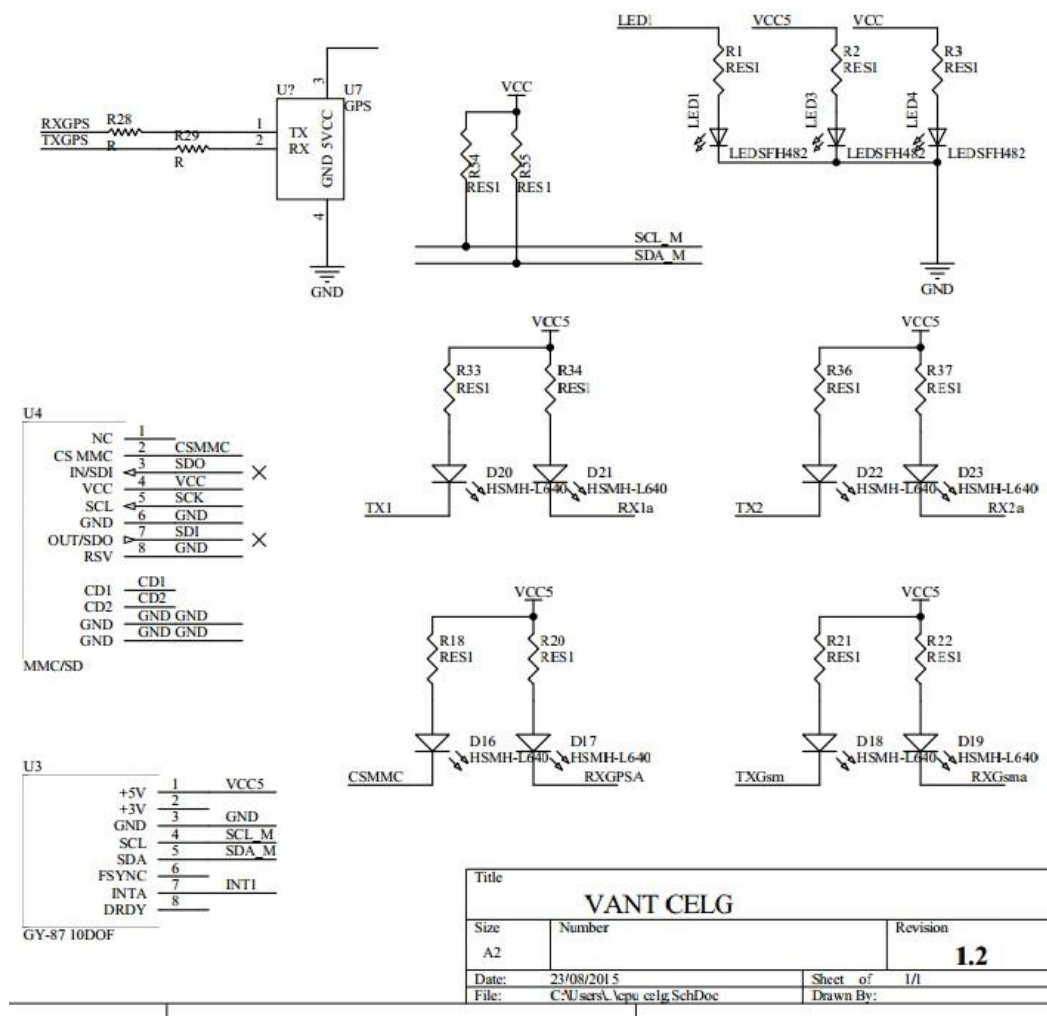


Figura 64 Leds de indicação e sensores.

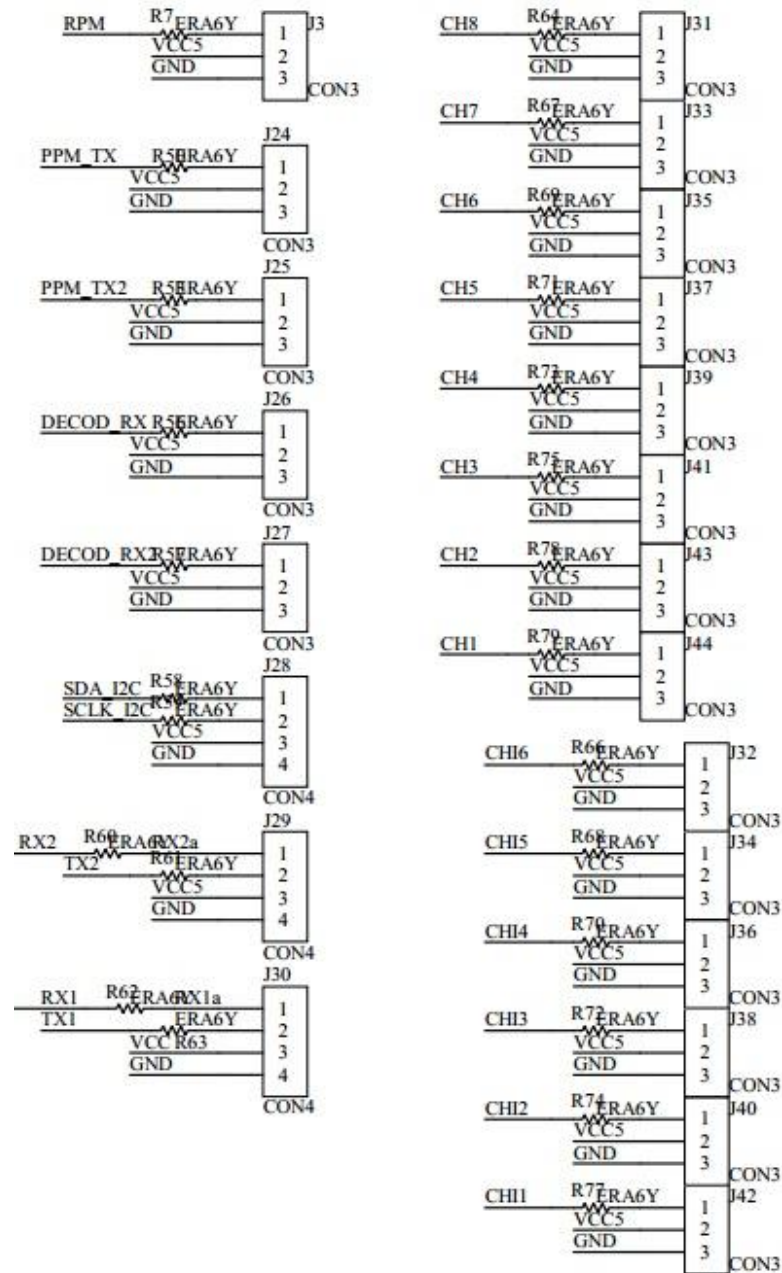


Figura 65 Conexões de servos e portas seriais.

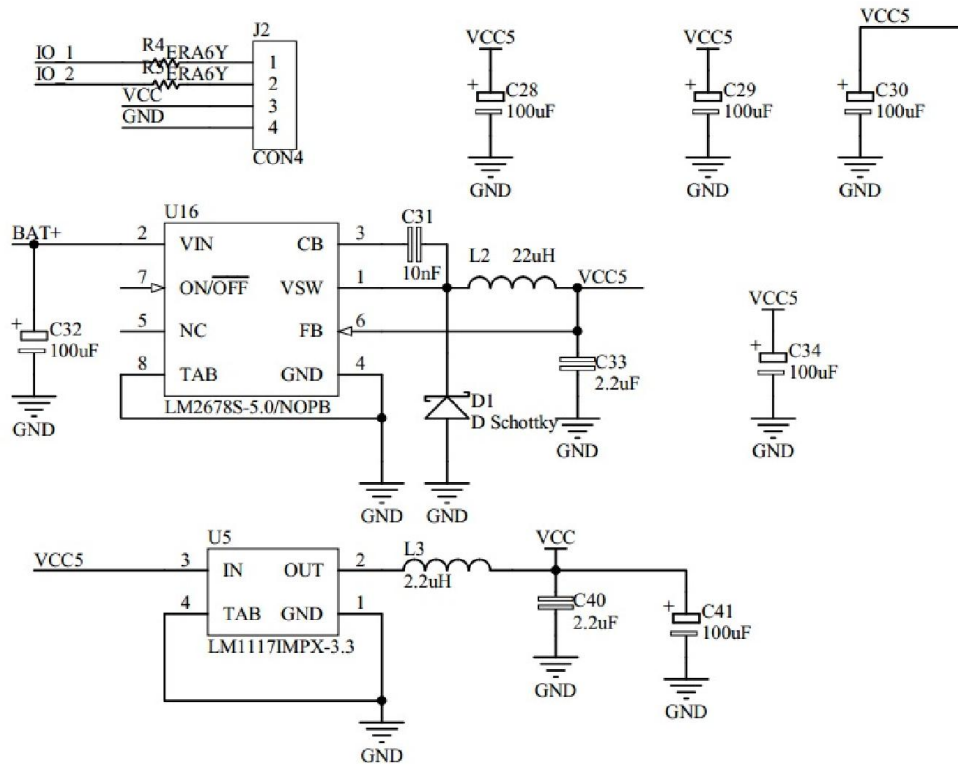


Figura 66 Fonte de alimentação

9.7 Função Polinomial

UINT16 CalculateCrc(UINT8 *data, UINT32 len)

```
{
    UINT i;
    UINT16 crc = 0;
    while(len--)
    {
        i = (crc >> 12) ^ (*data >> 4);
        crc = crc_table[i & 0x0F] ^ (crc << 4);
        i = (crc >> 12) ^ (*data >> 0);
        crc = crc_table[i & 0x0F] ^ (crc << 4);
        data++;
    }
    return (crc & 0xFFFF);
}
```

9.8 Código de inicialização

```
#include "Definicoes Globais.h"
#include "Config.h"
#include <libpic30.h>
#include <outcompare.h>
#include <pps.h>
#include <stdio.h>
#include <math.h>
#include <limits.h>
#include <stdlib.h>
#include <adc.h>
#include <InCap.h>

extern unsigned int  tempo_atualizador; // usado na atualizacao do status do modelo
extern unsigned long  T_status;        // usado no controle do status do
extern ADCA ADCBufferA;
extern ADCB ADCBufferB;

void configura_micro(void);
void config_OUT_SERVO(void);
void config_IN_RX(void);
void configura_TIMER(void);
void configura_ADC(void);
void cfgUart(void);
void configura_PORTAS(void);

void config_OUT_SERVO(void){

CloseOC1();
ConfigIntOC1(OC_INT_OFF & OC_INT_PRIOR_5);
OpenOC1(OC_IDLE_CON & OC_TIMER5_SRC & OC_CONTINUE_PULSE ,0, 65000, 7500);
OC1CON2bits.SYNCSEL=0x1f;
OC1CON2bits.OCINV=0x1;
```

```

CloseOC2();
ConfigIntOC2(OC_INT_OFF & OC_INT_PRIOR_5);
OpenOC2(OC_IDLE_CON & OC_TIMER5_SRC & OC_CONTINUE_PULSE ,0, 65000, 7500);
OC2CON2bits.SYNCSEL=0x1f;
OC2CON2bits.OCINV=0x1;

```

```

CloseOC3();
ConfigIntOC3(OC_INT_OFF & OC_INT_PRIOR_5);
OpenOC3(OC_IDLE_CON & OC_TIMER5_SRC & OC_CONTINUE_PULSE ,0, 65000, 7500);
OC3CON2bits.SYNCSEL=0x1f;
OC3CON2bits.OCINV=0x1;

```

```

CloseOC4();
ConfigIntOC4(OC_INT_OFF & OC_INT_PRIOR_5);
OpenOC4(OC_IDLE_CON & OC_TIMER5_SRC & OC_CONTINUE_PULSE ,0, 65000, 7500);
OC4CON2bits.SYNCSEL=0x1f;
OC4CON2bits.OCINV=0x1;

```

```

CloseOC8();
IFS2bits.OC8IF = 0;
ConfigIntOC8(OC_INT_ON & OC_INT_PRIOR_2);
OpenOC8(OC_IDLE_CON & OC_TIMER5_SRC & OC_CONTINUE_PULSE ,0, 5000, 500);
OC8CON2bits.SYNCSEL=0x1f;
OC8CON2bits.OCINV=0x1;

```

```

Canal[1]=500;

```

```

Canal[2]=500;

```

```

Canal[3]=500;

```

```

Canal[4]=500;

```

```

Canal[5]=00;

```

```

Canal[6]=0;

```

```

Canal[7]=1000;

```

```

Canal[8]=500;

```

```

Canal[9]=500;

```

```

Canal[10]=500;

```

```

Canal[11]=500;

```

```

RXin1[1]=500;

```

```

RXin1[2]=500;
RXin1[3]=500;
RXin1[4]=500;
RXin2[1]=500;
RXin2[2]=500;
RXin2[3]=500;
RXin2[4]=500;
}

/* *****configuracao para um rx PPM usa 2 IC em
cascata***** */

void config_IN_RX(void){
    IC1CON1bits.ICBNE=0;    /*apenas leitura indica capturas no buffer*/
    IC1CON1bits.ICOV=0;     /*No input capture overflow occurred*/
    IC1CON1bits.ICI=0;      /*Interrupt para qualquer evento*/
    IC1CON1bits.ICM=1;      /*captura eventos na borda de subida e descida*/
    IC1CON1bits.ICTSEL=2;   /*TMR4 como fonte de contagem*/
    IC1CON1bits.ICSIDL=0;   /*Input capture module will continue to operate in CPU Idle mode*/
    IC1CON2bits.IC32 = 0;   // Cascade module operation is disabled
    IC1CON2bits.ICTRIG = 0; // Input source used to synchronize the input capture timer of
    IC1CON2bits.TRIGSTAT = 0; // IC1TMR has not been triggered and is being held clear
    IC1CON2bits.SYNCSEL = 0x1f; // No Sync or Trigger source for the IC1 module
    IFS0bits.IC1IF = 0;    /*Zera interrupcao IC1*/
    IEC0bits.IC1IE = 1;    /*Set the IC7 interrupt enable bit */
    IC2CON1bits.ICOV=0;     /*No input capture overflow occurred*/
    IC2CON1bits.ICI=0;      /*Interrupt para qualquer evento*/
    IC2CON1bits.ICTSEL=2;   /*TMR4 como fonte de contagem*/
    /*ICTSEL<12:10>: Input Capture Timer Select bits
    111 = Peripheral clock (FP) is the clock source for the ICx module
    110 = Reserved
    101 = Reserved
    4 100 = Timer1 is the clock source for the ICx module
    3 011 = Timer5 is the clock source for the ICx module
    2 010 = Timer4 is the clock source for the ICx module
    1 001 = Timer2 is the clock source for the ICx module

```

```

    0 000 = Timer3 is the clock source for the ICx module
    */

    IC2CON1bits.ICSIDL=0;    /*Input capture module will continue to operate in CPU Idle mode*/
    IC2CON2bits.IC32 = 0;    // Cascade module operation is disabled
    IC2CON2bits.ICTRIG = 0;  // Input source used to synchronize the input capture timer of
    IC2CON2bits.TRIGSTAT = 0; // IC1TMR has not been triggered and is being held clear
    IC2CON2bits.SYNCSEL = 0x1f; // No Sync or Trigger source for the IC1 module
    IC2CON1bits.ICM=3;        /*Capture evento de borda de subida */
    IFS0bits.IC2IF = 0;      /*zera interrupcao IC2*/
    IEC0bits.IC2IE = 1;
    IPC1bits.IC2IP=1;

    IC3CON1=IC4CON1=IC7CON1=IC8CON1=IC9CON1=IC10CON1=IC1CON1;
    IC3CON2=IC4CON2=IC7CON2=IC8CON2=IC9CON2=IC10CON2=IC1CON2;
    IFS2bits.IC3IF = 0;      /*Zera interrupcao IC1*/
    IEC2bits.IC3IE = 1;      /*Set the IC7 interrupt enable bit */
    IFS2bits.IC4IF = 0;      /*Zera interrupcao IC1*/
    IEC2bits.IC4IE = 1;      /*Set the IC7 interrupt enable bit */
    IFS1bits.IC7IF = 0;      /*Zera interrupcao IC1*/
    IEC1bits.IC7IE = 1;      /*Set the IC7 interrupt enable bit */
    IFS1bits.IC8IF = 0;      /*Zera interrupcao IC1*/
    IEC1bits.IC8IE = 1;      /*Set the IC7 interrupt enable bit */
    IFS5bits.IC9IF = 0;      /*Zera interrupcao IC1*/
    IEC5bits.IC9IE = 1;      /*Set the IC7 interrupt enable bit */
    IFS7bits.IC10IF = 0;     /*Zera interrupcao IC1*/
    IEC7bits.IC10IE = 1;     /*Set the IC7 interrupt enable bit */
}

void configura_micro(void){
    PLLFBD=43;                // DEFINIDO NO CONFIG
    CLKDIVbits.PLLPOST=0;     // N1=2
    CLKDIVbits.PLLPRE=0;      // N2=2
    CLKDIV=0;
    // Disable Watch Dog Timer
    RCONbits.SWDTEN=0;
    __delay_ms(200);
}

```

```

#ifdef Debug
    __C30_UART=1;
    //U2BRG = 38;
    //U2MODEbits.UARTEN = 1;
#endif

    // Wait for PLL to lock
    while(OSCCONbits.LOCK!=1) {};
    __delay_ms(200);
    //ctglPinInit();
}

void configura_TIMER(void){
    T1CON = 0;           // Timer reset
    IFS0bits.T1IF = 0;   // Reset Timer1 interrupt flag
    IPC0bits.T1IP = 6;    // Timer1 Interrupt priority level=4
    IEC0bits.T1IE = 1;    // Enable Timer1 interrupt
    T1CONbits.TCKPS0=1;
    TMR1= (int)FOS/100;
    PR1 = (int)((FCY/8)/1000); //T1PRIOD; //0xc350;      // Timer1 period register = ?????
    T1CONbits.TON = 1;    // Enable Timer1 and start the counter
    T4CON = 0;           // Timer reset
    IFS1bits.T4IF = 0;    // Reset Timer4 interrupt flag
    IPC6bits.T4IP = 2;    // Timer1 Interrupt priority level=4
    IEC1bits.T4IE = 0;    // Enable Timer4 interrupt
    T4CONbits.TCKPS0=0;
    T4CONbits.TCKPS1=1;
    TMR4= (int)FOS/100;
    PR4 = (int)((FCY/8)/100); //T1PRIOD; //0xc350;      // Timer1 period register = ?????
    T4CONbits.TON = 1;    // Enable Timer1 and start the counter
    T5CON = 0;           // Timer reset
    IFS1bits.T5IF = 0;    // Reset Timer1 interrupt flag
    IPC7bits.T5IP = 6;    // Timer1 Interrupt priority level=4
    IEC1bits.T5IE = 1;    // Enable Timer1 interrupt
    T5CONbits.TCKPS0=1;
    T5CONbits.TCKPS1=0;
}

```

```

TMR5= (int)FOS/10000;
PR5 = 20000;//(int)((FCY/8)/1000)/200;
T5CONbits.TON = 1;    // Enable Timer1 and start the counter
//tempo_atualizador=Config.TEMP_STATUSMODELO;
}

void configura_ADC(void){
    ANSELB = ANSELC = ANSELD = ANSELE = ANSELG = 0x0000;
    ANSELBbits.ANSB11=0;
    ANSELEbits.ANSE5=1;
    ANSELEbits.ANSE6=1;
    ANSELBbits.ANSB12=1;
    ANSELBbits.ANSB14=1;
    TRISEbits.TRISE5=1;
    TRISEbits.TRISE6=1;
    TRISBbits.TRISB12=1;
    TRISBbits.TRISB14=1;
    TRISBbits.TRISB7=1;
    TRISBbits.TRISB11=1;
    AD1CON1bits.FORM = 0;    // Data Output Format: Signed Fraction (Q15 format)
    AD1CON1bits.SSRC = 7;    // Sample Clock Source: GP Timer starts conversion
    AD1CON1bits.ASAM = 1;    // Sample Control: Sampling begins immediately after conversion
    AD1CON1bits.AD12B = 1;    // 10-bit ADC operation
    AD1CON2bits.CSCNA = 1;    // Scan Input Selections for CH0+ during Sample A bit
    AD1CON2bits.CHPS = 0;    // Converts CH0
    AD1CON3bits.ADRC = 0;    // Clock is derived from Systems Clock
    AD1CON3bits.ADCS = 63;    // Clock Tad=Tcy*(ADCS+1)= (1/40M)*64 = 1.6us (625Khz)
    AD1CON3bits.SAMC =31;    // ADC Conversion Time for 10-bit Tc=12*Tab = 19.2us
    AD1CON1bits.ADDMABM = 0; // DMA buffers are built in scatter/gather mode
    AD1CON2bits.SMPI = 3;    // numero de canais ADC que setrao escaneados
    AD1CON4bits.DMABL = 3;    // Each buffer contains 8 words
    AD1CON4bits.ADDMAEN = 0; // Conversion results stored in ADCxBUF0 register
    AD1CON2bits.ALTS=0;
    AD1CSSH = 0x0000;
    AD1CSSL = 0x0000;
    AD1CSSHbits.CSS30 = 1;

```

```

AD1CSSHbits.CSS29 = 1;
AD1CSSLbits.CSS12 = 1;
AD1CSSLbits.CSS14 = 1; // Scan AIN0, AIN1, AIN2, AIN3 inputs
IFS0bits.AD1IF = 0;      // Clear the A/D interrupt flag bit
IEC0bits.AD1IE = 1;      // Do Not Enable A/D interrupt
AD1CON1bits.ADON = 1;    // Turn on the A/D converter
}

void cfgUart(void){
    U1MODEbits.STSEL = 0;      // 1-stop bit
    U1MODEbits.PDSEL = 0;      // No Parity, 8-data bits
    U1MODEbits.ABAUD = 0;      // Autobaud Disabled
    U1MODEbits.BRGH = 0;      // Autobaud Disabled
    U1BRG = BRGVAL1;          // BAUD Rate Setting for 9600
    U2MODEbits.STSEL = 0;      // 1-stop bit
    U2MODEbits.PDSEL = 0;      // No Parity, 8-data bits
    U2MODEbits.ABAUD = 0;      // Autobaud Disabled
    U2BRG = BRGVAL2;          // BAUD Rate Setting for 9600
    U3MODEbits.STSEL = 0;      // 1-stop bit
    U3MODEbits.PDSEL = 0;      // No Parity, 8-data bits
    U3MODEbits.ABAUD = 0;      // Autobaud Disabled
    U3BRG = BRGVAL3;          // BAUD Rate Setting for 9600
    U4MODEbits.STSEL = 0;      // 1-stop bit
    U4MODEbits.PDSEL = 0;      // No Parity, 8-data bits
    U4MODEbits.ABAUD = 0;      // Autobaud Disabled
    U4BRG = BRGVAL4;          // BAUD Rate Setting for 9600
    /*******
    // STEP 1:
    // Configure UART for DMA transfers
    /*******
    U1STAbits.UTXISEL0 = 0;    // Interrupt after one Tx character is transmitted
    U1STAbits.UTXISEL1 = 0;
    U1STAbits.URXISEL = 0;      // Interrupt after one RX character is received
    U1STAbits.URXISEL = 3; // interrupcao apenas quando recebe 4 caracteres tem que ser 1 para o
    simulador
    IFS0bits.U1RXIF = 0;      //

```



```

IEC0bits.U1RXIE  = 1;
IFS0bits.U1TXIF  =0;
IEC0bits.U1TXIE  =0;
U2STAbits.UTXISEL0 = 0;          // Interrupt after one Tx character is transmitted
U2STAbits.UTXISEL1 = 0;
U2STAbits.URXISEL  = 0;          // Interrupt after one RX character is received
#ifdef SIMULADOR
    U2STAbits.URXISEL  = 1;      // interrupcao apenas quando recebe 1 caracteres
#else
    U2STAbits.URXISEL  = 3;      // interrupcao apenas quando recebe 4 caracteres
#endif
IFS1bits.U2RXIF  = 0;          //
IEC1bits.U2RXIE  = 1;
U3STAbits.UTXISEL0 = 0;          // Interrupt after one Tx character is transmitted
U3STAbits.UTXISEL1 = 0;
U3STAbits.URXISEL  = 0;          // Interrupt after one RX character is received
U3STAbits.URXISEL  = 1;          // interrupcao apenas quando recebe 4 caracteres
IFS5bits.U3RXIF  = 0;          //
IEC5bits.U3RXIE  = 1;
U4STAbits.UTXISEL0 = 0;          // Interrupt after one Tx character is transmitted
U4STAbits.UTXISEL1 = 0;
U4STAbits.URXISEL  = 0;          // Interrupt after one RX character is received
U4STAbits.URXISEL  = 3;          // interrupcao apenas quando recebe 4 caracteres
IFS5bits.U4RXIF  = 0;          //
IEC5bits.U4RXIE  = 1;
/*****
// STEP 2:
// Enable UART Rx and Tx
*****/
U1MODEbits.UARTEN = 1;          // Enable UART
U1STAbits.UTXEN   = 1;          // Enable UART Tx
U2MODEbits.UARTEN = 1;          // Enable UART
U2STAbits.UTXEN   = 1;          // Enable UART Tx
U3MODEbits.UARTEN = 1;          // Enable UART
U3STAbits.UTXEN   = 1;          // Enable UART Tx
U4MODEbits.UARTEN = 1;          // Enable UART

```

```

U4STAbits.UTXEN  = 1;          // Enable UART Tx
}

void configura_PORTAS(void)
{
PPSUnLock;
PPSInput(PPS_U1RX, PPS_RP80); // Maps RX1 input to RP80 pin
PPSOutput(OUT_FN_PPS_U1TX, OUT_PIN_PPS_RP97); // Maps TX1 output to RP97
PPSInput(PPS_U2RX, PPS_RP68); // Maps RX2 input to RP70 pin
PPSOutput(OUT_FN_PPS_U2TX, OUT_PIN_PPS_RP67); // Maps TX2 output to RP69
PPSInput(PPS_U3RX, PPS_RP70); // Maps RX3 input to RP68 pin
PPSOutput(OUT_FN_PPS_U3TX, OUT_PIN_PPS_RP69); // Maps TX3 output to RP67
PPSInput(PPS_U4RX, PPS_RP61); // Maps RX4 input to RP61 pin
PPSOutput(OUT_FN_PPS_U4TX, OUT_PIN_PPS_RP64); // Maps TX4 output to RP64
PPSInput(PPS_IC2, PPS_RP42); // Maps RX1 input to RP42 pin      //RX ENTRADA PPM2
PPSInput(PPS_IC3, PPS_RP41); // Maps RXCN1 input to RP40 pin      //cn1
PPSInput(PPS_IC4, PPS_RP40); // Maps RXCN2 input to RP40 pin      //cn2
PPSInput(PPS_IC7, PPS_RP39); // Maps RXCN2 input to RP40 pin      //cn3
PPSInput(PPS_IC8, PPS_RP38); // Maps RXCN2 input to RP40 pin      //cn4
PPSInput(PPS_IC9, PPS_RP134); // Maps RXCN2 input to RP40 pin      //cn5
PPSInput(PPS_IC10, PPS_RP35); // Maps RXCN2 input to RP40 pin      //cn6
PPSInput(PPS_SDI1, PPS_RP99);          // Maps RX4 input to RP20 pin
PPSOutput(OUT_FN_PPS_SDO1, OUT_PIN_PPS_RP98); // Maps TX4 output to RP64
PPSOutput(OUT_FN_PPS_SCK1, OUT_PIN_PPS_RP102); // Maps TX4 output to RP64
PPSOutput(OUT_FN_PPS_OC1, OUT_PIN_PPS_RP120);
PPSOutput(OUT_FN_PPS_OC2, OUT_PIN_PPS_RP118);
PPSOutput(OUT_FN_PPS_OC3, OUT_PIN_PPS_RP87);
PPSOutput(OUT_FN_PPS_OC4, OUT_PIN_PPS_RP84);
PPSOutput(OUT_FN_PPS_SDO1, OUT_PIN_PPS_RP71); // Maps SPI1 output to RP71*/
PPSOutput(OUT_FN_PPS_OC8, OUT_PIN_PPS_RP82);
PPSLock;
IEC4bits.U2EIE = 0;
}

```