

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

nota final
8,0 (nota)
Hm

GABRIEL SEVERINO DA SILVA JÚNIOR

Sistema de aquisição de dados embarcado em aeronave

São Paulo
2005

GABRIEL SEVERINO DA SILVA JÚNIOR

Sistema de aquisição de dados embarcado em aeronave

892

Projeto de conclusão de curso apresentado à Escola
Politécnica da Universidade de São Paulo para
obtenção do título de Engenheiro Mecânico

Área de Concentração: Automação e Controle
Orientador: Dr. Antonio Luís de Campos Mariani

São Paulo
2005

TF_05
Si 385

DEDALUS - Acervo - EPMN



31600011872

2305785

Aos meus pais,

Por me educarem com muito esforço,

Sempre confiantes no meu potencial,

E a minha filha, por ser a razão do meu viver,

E a dona do meu Eterno Amor.

AGRADECIMENTOS

Este trabalho foi concluído com muito esforço e dedicação. Porém, sem a ajuda e a presença de pessoas especiais em minha vida, a conclusão dele seria impossível. Agradeço:

Aos meus pais, Gabriel e Sueli, por me educarem sempre confiantes em meu potencial, e por me criarem com grande dedicação. Tudo que sou e tenho, devo a eles e ao Senhor.

Aos meus irmãos e madrasta Ricardo, Marcelo, Amanda e Isabel por sempre me apoiarem, especialmente nos momentos de fraqueza.

Ao Renato Diogo, à Beatrice, ao Ueli, ao Fernando D'Agosto, ao Luciano Eizak, ao Danilo e Pedro Grassi, a Patrícia, e ao Rafael Marconato por serem grandes amigos. Ao Albert Nissimoff, ao Pedro, e ao Luiz Golin, por ajudarem muito com este trabalho. Ao Giovani Amianti, ao Marco Ceze, e ao Fábio Mazza, pelo incentivo e ajuda com meu desenvolvimento acadêmico e pessoal durante dois bons anos de participação no SAE Aerodesign.

Às equipes Aclive e PoliAclive TAM pelos bons dias de trabalho, ensaios, vôos, e competição SAE Aerodesign.

Um agradecimento especial à Dilcelli Soares, por ser uma das pessoas que mais me ajudou, e mais esteve presente em todos os momentos.

Ao Professor Orientador Dr. Antônio Luis de Campos Mariani, por me orientar de maneira excelente nesse trabalho, mas principalmente por ser um grande exemplo de pessoa, tanto no lado profissional quanto no pessoal.

Ao Fabrício Honda, Sílvio Steinberg e André Fregolente por confiarem na minha capacidade profissional em um momento crucial.

Aos meus sogros de consideração, Dona Darci e Sr. João, por cuidarem da minha filha com muito amor, e por me acatarem sempre tão bem.

Aos colaboradores e parceiros: Freescale, e Equipe PoliAclive TAM.

Ao Prof. Rui Baptista e à *BLUE TEC* pela ajuda em um momento decisivo.

A minha filha, Princesinha Mariana, dona do meu Eterno Amor e da Razão do meu Viver.

À Deus, por ser a base rígida que me impede de cair.

E a todas aquelas pessoas, que direta ou indiretamente me ajudaram e rezaram por mim.

*"Once upon a midnight dreary, while I pondered, weak and weary,
Over many a quaint and curious volume of forgotten lore,
While I nodded, nearly napping, suddenly there came a tapping,
As of some one gently rapping, rapping at my chamber door.
"Tis some visitor", I muttered, "tapping at my chamber door.
Only this, and nothing more".(...)"*

The Raven, Edgar Allan Poe (1845)

Sistema de aquisição de dados embarcado em aeronave

Gabriel Severino da Silva Júnior
Antonio Luís de Campos Mariani, Prof. Orientador

Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos

Palavras-chave: Sistema Embarcado, Aeronave, Avionica, UAV, Aquisição de Dados

RESUMO

O presente trabalho trata do projeto, da manufatura e de testes de um sistema de aquisição de dados a ser embarcado em aeronave, focando tanto na arquitetura de hardware como de software. O enfoque adotado tem caráter prático, com a finalidade de, em aplicações futuras, estimar as equações que regem o movimento de corpo rígido das aeronaves teste, visando à implementação de um sistema de controle e navegação das mesmas.

As aeronaves teste rádio controladas foram desenvolvidas pela equipe PoliActive Tam de Aerodesign. Estas aeronaves são descritas como aeromodelos cargueiros de médio porte, e têm como missão carregar a maior carga útil possível, com restrições de pista, motor e dimensões de asa e fuselagem.

O objetivo deste trabalho é projetar, manufaturar e testar uma arquitetura embarcada que execute a leitura e armazenamento dos sinais de todos os sensores abordo da aeronave, e que seja expansível. Os sensores para medição de pressão atmosférica, pressão dinâmica, umidade relativa, temperatura atmosférica, ângulos de ataque longitudinal e lateral, e rotação da hélice também serão selecionados ao longo deste projeto. Os dados amostrados serão futuramente utilizados para verificação e modificação da estimativa do modelo dinâmico das aeronaves.

A configuração proposta é modular, baseada em microcontroladores. Desta forma, o presente projeto pode ser definido como o módulo de aquisição de sinais analógicos. Este módulo será projetado de modo a garantir sua integração com outros módulos.

Um extenso estudo de “estado da arte” foi realizado e também será apresentado. Serão ilustrados nesta análise diversos projetos realizados ao redor do globo, e suas principais características.

***Apresentado na disciplina PMR 2550 - Trabalho de Conclusão de Curso II
Dezembro de 2005.***

Coordenadores Prof. Lucas A . Moscato e Prof. Edson Gomes

Aircraft Embedded Data Acquisition System

Gabriel Severino da Silva Júnior
Antonio Luís de Campos Mariani, Supervisor Prof.

Escola Politécnica da Universidade de São Paulo
Department of Mechatronics Engineering and Mechanical Systems

Key words: Embedded System, Aircraft, Avionics, UAV, Data Acquisition

ABSTRACT

This paper deals with the design, manufacture, and tests of an aircraft embedded data acquisition system, focusing on both hardware and software architectures. A practical approach is adopted, aiming in future applications the test aircraft dynamics assessment, and the design of a control and navigation system.

The radio controlled test aircrafts were designed by Aerodesign team PoliActive TAM. These aircrafts are defined as middle size cargo air models, and their mission is to carry as much payload as possible, with restriction of runway, wingspan, propulsion, and fuselage size.

The goal of this paper is to design, manufacture and test an embedded architecture that might read and store all sensors signals onboard the air models, and might be expansible. The barometric pressure, dynamic pressure, temperature, relative humidity, lateral and longitudinal angles of attack, and the motor rotation speed sensors will also be selected along this paper. The sampled data will be used in future researches to verify and modify the estimated aircraft dynamics.

The proposed design is based on modular micro controlled architecture. This way, this paper may be defined as the design of an analog data acquisition module. This module will be designed aiming its future integration with other modules.

An extended study of "the state the art" was made, and will also be presented. Will be shown in this analysis many projects designed around the globe, and their main characteristics.

***Presented to PMR 2550 - Trabalho de Conclusão de Curso II
December 2005.***

Professors Dr. Lucas A . Moscato and Dr. Edson Gomes

LISTA DE FIGURAS

Figura 2.1 – Grandezas mensuradas e estimadas	21
Figura 2.2 – Esboço da placa de aquisição	21
Figura 2.3 – Projeto de 2004	22
Figura 2.4 – Teste em vôo, velocidade por tempo	22
Figura 2.5 – Desenho em CAD da aeronave	23
Figura 2.6 – Foto da aeronave	24
Figura 3.1 - Sistema de eixos	25
Figura 3.2 – Movimento da aeronave	26
Figura 3.3 – Ângulos de Euler	27
Figura 4.1 – ARMAR	33
Figura 4.2 – Arquitetura embarcada no ARMAR	33
Figura 4.3 – OLGA	34
Figura 4.4 – Arquitetura do sistema de controle OLGA	34
Figura 4.5 – GenBase II	35
Figura 4.6 – Arquitetura embarcada no GenBase II	35
Figura 4.7 – Hardware e plataforma	36
Figura 4.8 – <i>Navigator</i>	37
Figura 4.9 – Sistema embarcado no <i>Navigator</i>	37
Figura 4.10 – Aeronaves teste e PAD	38

Figura 4.11 – Sistema embarcado nos UAVs, nos PADs, e da estação em solo	38
Figura 4.12 – Plataforma experimental	39
Figura 4.13 – Diagrama de blocos do circuito eletrônico	39
Figura 4.14 – Tethra	40
Figura 4.15 – Sistema eletrônico do Tethra	40
Figura 4.16 – <i>Smart Car</i>	41
Figura 4.17 – <i>DragonFly</i>	42
Figura 4.18 – Eletrônica embarcada	42
Figura 4.19 – Esquema eletrônico do <i>Dragon Fly</i>	42
Figura 4.20 – Algoritmo para tratamento do sinal de vídeo	43
Figura 4.21 – Plataforma de testes	43
Figura 4.22 – Arquitetura de <i>software</i>	43
Figura 4.23 – Plataformas de testes	44
Figura 4.24 – Eletrônica embarcada (60,4 g)	44
Figura 4.25 – Diagramas de bloco da eletrônica	44
Figura 4.26 – UAV GTmax	45
Figura 4.27 – Aviônica	45
Figura 4.28 – Diagrama de blocos do sistema	45
Figura 4.29 – <i>Piper J3</i>	46
Figura 4.30 – Diagrama de blocos do sistema	46
Figura 4.31 – <i>Mini Child</i>	47
Figura 4.32 – Aviônica	47

Figura 4.11 – Sistema embarcado nos UAVs, nos PADs, e da estação em solo	38
Figura 4.12 – Plataforma experimental	39
Figura 4.13 – Diagrama de blocos do circuito eletrônico	39
Figura 4.14 – Tethra	40
Figura 4.15 – Sistema eletrônico do Tethra	40
Figura 4.16 – <i>Smart Car</i>	41
Figura 4.17 – <i>DragonFly</i>	42
Figura 4.18 – Eletrônica embarcada	42
Figura 4.19 – Esquema eletrônico do <i>Dragon Fly</i>	42
Figura 4.20 – Algoritmo para tratamento do sinal de vídeo	43
Figura 4.21 – Plataforma de testes	43
Figura 4.22 – Arquitetura de <i>software</i>	43
Figura 4.23 – Plataformas de testes	44
Figura 4.24 – Eletrônica embarcada (60,4 g)	44
Figura 4.25 – Diagramas de bloco da eletrônica	44
Figura 4.26 – UAV GTmax	45
Figura 4.27 – Aviônica	45
Figura 4.28 – Diagrama de blocos do sistema	45
Figura 4.29 – <i>Piper J3</i>	46
Figura 4.30 – Diagrama de blocos do sistema	46
Figura 4.31 – <i>Mini Child</i>	47
Figura 4.32 – Aviônica	47

Figura 4.33 – Eletrônica embarcada no <i>Mini Child</i>	47
Figura 4.34 – ARARA	48
Figura 4.35 – Hardware	48
Figura 5.1 – Metodologia adotada no projeto do <i>hardware</i>	50
Figura 5.2 – Curva de descarga, bateria Li-MnO ₂	52
Figura 5.3 – Curva de descarga, pilha Ni-MH	52
Figura 5.4 – Faixa de tensão dos níveis lógicos dos principais padrões	55
Figura 5.5 – Sensores de pressão Freescale, MPXH6101A6U (esq.) e MPXV5004DP	57
Figura 5.6 – Sensores de umidade Honeywell, HIH-3610-002 (esq.) e HIH-3610-001	57
Figura 5.7 – Representação do sensor LM335	58
Figura 5.8 – Foto diodo IR BPV10NF (esq.) e LED IR TSUS4300	60
Figura 5.9 – Microcontrolador com encapsulamento PDIP de 40 pinos	68
Figura 5.10 – Circuito para sensor de pressão diferencial (analógico)	70
Figura 5.11 – Circuito para sensor de pressão barométrica absoluta (analógico)	70
Figura 5.12 – Circuito para sensor de umidade (analógico)	71
Figura 5.13 – Circuito para sensor de temperatura (analógico)	71
Figura 5.14 – Circuito para sensor de ângulo de ataque longitudinal e lateral	71
Figura 5.15 – Medidor de rotação	72
Figura 5.16 – Circuito para o MMC	73
Figura 5.17 – Circuito para fonte de 5V	74
Figura 5.18 – Circuito para fonte de 3,3V	74
Figura 5.19 – Circuito para comunicação serial com PC	75

Figura 5.20 – Circuito para comunicação paralela, visando futura expansão do sistema	75
Figura 5.21 – Circuito do microcontrolador	77
Figura 5.22 – Circuito modificado para fonte de 5V	79
Figura 6.1 – Diagrama de estados	83
Figura 6.2 – Leitura de um byte na memória I2C	86
Figura 6.3 – Escrita de um byte na memória I2C	86
Figura 6.4 – Leitura de um byte no MMC	87
Figura 6.5 – Escrita de um byte no MMC	87
Figura 6.6 – Circuito programador JDM para PIC16F877A	90
Figura 7.1 – Circuito teste em funcionamento	91
Figura 7.2 – Protótipo final	93

LISTA DE TABELAS

Tabela 2.1 – Desempenho	23
Tabela 2.2 – Características geométricas	24
Tabela 3.1 – Pequenas perturbações	30
Tabela 3.2 – Linearização do diferencial das forças e dos momentos	30
Tabela 4.1 – Principais empresas de UAVs e aviação	49
Tabela 5.1 – Sensores de pressão	56
Tabela 5.2 – Sensores de temperatura	58
Tabela 5.3 – Microcontroladores	68
Tabela 5.4 – Corrente drenada e potência dissipada	78
Tabela 5.5 – Materiais e orçamento final	80
Tabela 6.1 – Registradores configurados	84
Tabela 6.2 – Pinos utilizados na programação	88
Tabela 6.3 – Comandos no modo de programação	89

SUMÁRIO

1. Introdução	15
1.1. Histórico de UAVs	16
1.2. Projeto de UAVs	17
1.3. Objetivo	18
1.4. Organização do Texto	19
2. Escopo e Aeronaves Teste	20
2.1. Escopo	20
2.2. Projeto de Aquisição de Dados de 2004	22
2.3. Aeronaves Teste	23
3. Modelagem da Dinâmica de Aeronaves	25
3.1. Referenciais e Nomenclatura	24
3.2. Modelagem	26
3.2.1. Modelo Dinâmico Não Linear	27
3.2.2. Modelo Dinâmico Linear	29
4. Estado da Arte	32
4.1. Projeto de Sistemas Embarcados	33
4.1.1. ARMAR	33
4.1.2. OLGA	34

4.1.3. GenBase II	35
4.1.4. Sistema de Controle para Múltiplos Veículos Autônomos	36
4.1.5. <i>Navigator</i>	37
4.1.6. Sistema de Reconhecimento Aéreo Autônomo	38
4.1.7. Plataforma Experimental e Controle de Helicóptero	39
4.1.8. Tethra	40
4.1.9. <i>Smart Car</i>	41
4.2. Projeto de Aeronaves Autônomas	41
4.2.1. <i>Dragon Fly</i>	42
4.2.2. Projeto Piloto de UAV	43
4.2.3. Mini UAVs	44
4.2.4. Sistema de Controle Adaptativo para UAVs	45
4.2.5. <i>Piper J3</i>	46
4.2.6. <i>Mini Child</i>	47
4.2.7. ARARA	48
4.3. Empresas e Produtos Comercializados	49
5. Projeto e Especificações do <i>Hardware</i> Embarcado	50
5.1. Requisitos e Especificações	50
5.2. Escolha de Componentes	54
5.2.1. Sensores	56
5.2.2. Circuitos integrados	62
5.2.3. Memória	63
5.2.4. Microcontrolador	67

5.3.	Projeto Esquemático	69
5.3.1.	Circuitos para os Sensores	70
5.3.2.	Circuito para o MMC	73
5.3.3.	Circuito de Alimentação	74
5.3.4.	Circuito para Comunicação Externa	75
5.3.5.	Circuito para o Microcontrolador	76
5.4.	Análise	77
5.5.	Lista de Materiais e Orçamento Final	80
6.	Projeto do <i>Firmware</i>	81
6.1.	Diagrama de Estados	81
6.2.	Configuração do PIC	83
6.3.	Descrição do <i>firmware</i>	85
6.4.	Programação do PIC	88
7.	Ensaio e Testes de Protótipos	91
8.	Conclusões	94

REFERÊNCIAS

ANEXO A – Sensores

ANEXO B – Circuitos integrados

ANEXO C – Multimedia Card SDM1-128 SanDisk e Memória 24LC04B Microchip

ANEXO D – Microcontrolador PIC16F877A Microchip

ANEXO E – Bibliotecas do compilador MikroC MikroElektronika e Código Fonte

Capítulo 1

Introdução

Sistemas embarcados estão presentes nos mais diversos campos de aplicação de engenharia. Alguns exemplos de aplicação são: aparelhos celular, PDAs, câmeras fotográficas, filmadoras, televisões, aparelhos de DVD, fornos microondas, máquinas de lavar, automóveis, aeronaves, etc.. O requisito principal em alguns desses casos é que o sistema responda em “Tempo Real”. Em poucas linhas “Tempo Real” significa, não somente alto desempenho, mas também tomadas de decisão devido às restrições de tempo, ou seja, as tarefas a serem executadas dependem do tempo em que os dados são recebidos. Aplicações de risco que necessitam de Sistemas de Tempo Real são denominados “*Hard Real Time Systems*”, ou em português, “Sistemas de Tempo Real Crítico”.

Sistemas embarcados em veículos aéreos possibilitam que estes veículos sejam utilizados como aeronaves de monitoração. Sistemas de tempo real embarcados em veículos aéreos têm a finalidade de transformá-los em veículos autônomos não tripulados (UAVs – *Unmanned Aerial Vehicles*). Estes veículos são objeto de estudo em vários países, sendo utilizados tanto em missões civis, como militares. Aplicações militares envolvem espionagem e reconhecimento em áreas de risco. Desde o final da década de 90, UAVs militares de alta performance vêm sendo utilizadas também como elementos-chave em estratégias militares. As aplicações civis envolvem monitoração de fronteiras, áreas urbanas e agrícolas, e observações atmosféricas.

1.1. Histórico de UAVs

O histórico das aeronaves não tripuladas (UAVs) inicia-se nos anos 50 com os EUA na frente de pesquisa de aeronaves de reconhecimento e espionagem não tripuladas. Na década de 60 os primeiros UAVs aparecem em conflitos armados com a missão de fazer reconhecimento sobre áreas inimigas fortemente defendidas, em que aeronaves convencionais eram abatidas com alto custo de vida e material. O primeiro esquadrão de veículos não tripulados foi estabelecido na Base Aérea Nellis, Nevada. Já nos anos 90 os EUA colocam em operação o *Predator* e o *Global Hawk*, principais UAVs militares em operação atualmente.

Em serviço desde junho de 1994, o primeiro modelo do *Predator* tinha autonomia de voo de 700 quilômetros, pesava 1,02 tonelada e media 8,22 metros, com envergadura de 14,8 metros. O modelo *Predator* RQ-1 desempenhou um papel importante nos Bálcãs, Iraque e Afeganistão. A experiência no Kosovo, mostrou que as indicações do GMTI (radar indicador de alvos em movimento na superfície) ampliaram o conhecimento do espaço de batalha, tornando os UAVs muito mais eficientes, eficazes e duráveis. O *Global Hawk* RQ-4A, operacional desde fevereiro de 1998, pode transmitir imagens precisas quase em tempo real. O modelo pesa 11,6 toneladas, mede 13,4 metros e tem 35 metros de envergadura. Em 2002, um *Predator*, da CIA, localizou e dizimou uma caravana que se movia no oeste do Afeganistão.

No Brasil, os UAVs são utilizados para coleta de dados na agricultura, em monitoração de fronteiras, monitoração de tempo, tráfico em favelas, etc. Atualmente, uma lei nacional permite o uso desses veículos somente por parte do exército, da marinha, da aeronáutica e da comunidade científica. O CTA, em parceria com a Embraer e a Avibrás, divulgou em nov/2005 que em 2006 realizará vãos das versões preliminares de seus veículos aéreos não tripulados.

1.2. Projeto de UAVs

O projeto de aeronaves autônomas não tripuladas (UAVs) pode ser dividido em 3 etapas:

1. Projeto da aeronave rádio controlada;
2. Projeto do sistema de sensoriamento;
3. Projeto do sistema de controle e navegação;

Na primeira etapa são definidos: a missão, os requisitos e as restrições, de modo que o projeto de aerodinâmica, estrutural, de acionamentos, de aterrissagem e de tração motora seja efetuado com sucesso.

Na segunda fase, todo o sistema embarcado deve ser projetado, desenvolvido e testado. Nesta etapa a arquitetura do software também é definida, e o algoritmo para aquisição dos dados já é implementando. Ao final desta fase a aeronave é definida como uma aeronave de sensoriamento rádio controlada. Caso haja transmissão de vídeo e dos sinais dos sensores para uma estação base, a denominação correta seria aeronave de monitoração remota rádio controlada. Caso o alcance da aeronave seja maior que o alcance da visão do piloto, e seja necessário implementar pilotagem remota utilizando-se transmissão de vídeo e dados de sensores, então a definição passa a ser aeronave de monitoração pilotada remotamente.

A terceira etapa consiste em projetar, desenvolver e testar o sistema autônomo de controle e navegação da aeronave. Ao final desta etapa a aeronave recebe a denominação de aeronave autônoma de sensoriamento ou aeronave autônoma de monitoração (remota ou não). A definição do tipo de aeronave pode também variar com a sua função (militar, civil, científica, etc.).

1.3. Objetivo

A Escola Politécnica da USP participa, desde 2001, da competição universitária denominada SAE Aerodesign, realizada em São José dos Campos. Para esta competição, as equipes projetam e fabricam aeronaves rádio controladas que devem obedecer às normas propostas pela SAE.

Considerando a metodologia de projeto de UAVs, apresentado no item 1.2, equivale a dizer que as equipes Poli Aerodesign da EPUSP cumprem a primeira etapa do projeto de um UAV. Portanto, o objetivo do autor é apresentar neste trabalho a segunda etapa do projeto de uma aeronave autônoma, ou seja, o projeto, o desenvolvimento e os testes do sistema embarcado, incluindo a arquitetura do software para leitura dos sensores.

O autor propõe o desenvolvimento de uma arquitetura modular microcontrolada para o sistema embarcado. O objetivo é projetar, manufaturar e testar, hardware, e software para futuro sensoriamento em voo e identificação da dinâmica das aeronaves teste. Para este trabalho, não será considerada a transmissão de vídeo nem de sinais de sensores via rádio frequência, ou seja, os dados amostrados serão lidos *offline* diretamente no sistema embarcado. Serão amostrados dados de sensores de temperatura, pressão, umidade relativa, ângulos de ataque lateral e longitudinal, e rotação da hélice.

1.4. Organização do Texto

O presente projeto está dividido em nove capítulos.

O segundo capítulo define de forma esquemática o objetivo deste trabalho, ilustrando as grandezas a serem medidas e estimadas, e esboçando o projeto. As aeronaves onde o sistema será embarcado também serão ilustradas nesta seção.

O terceiro capítulo apresenta a modelagem dinâmica e cinemática de corpos rígidos com seis graus de liberdade, focando obviamente em aeronaves.

O quarto capítulo faz a contextualização (estado da arte), introduzindo os conceitos de sistemas embarcados, e trazendo aplicações retiradas de uma revisão bibliográfica recente.

O quinto capítulo tem como objetivo mostrar detalhadamente o projeto e especificações do hardware embarcado.

O sexto capítulo detalha o projeto do software (*firmware*).

O sétimo capítulo descreve os testes e ensaios realizados, com o intuito de certificar a funcionalidade do sistema embarcado.

O oitavo e último capítulo apresenta as conclusões.

Capítulo 2

Escopo e Aeronaves Teste

Este capítulo apresenta o escopo deste trabalho, de forma esquemática, tratando também dos testes realizados com uma placa projetada em 2004. Também é objetivo deste item informar as características das aeronaves cujo sistema será embarcado.

Em futuras aplicações será estimado o comportamento dinâmico das aeronaves, portanto é necessário conhecer as características geométricas, aerodinâmicas, de tração, massa e inércia rotacional. Outras características, como pista para decolagem e pouso, e autonomia, também serão informadas aqui a título de divulgação do SAE Aerodesign, competição na qual os aeromodelos testes serão utilizados.

2.1. Escopo

Neste projeto serão utilizados sensores de ângulos de ataque longitudinal e lateral, de pressão (barométrica e dinâmica), de umidade relativa, de temperatura, e de rotação da hélice. Com os sinais destes sensores é possível (admitindo que são conhecidas: tração em função da velocidade relativa ao escoamento, a característica aerodinâmica da aeronave, os sinais de comando, e a velocidade em direção e módulo das perturbações atmosféricas com relação a um referencial supostamente inercial) estimar o histórico no tempo da velocidade da aeronave, o seu

arrasto e sustentação em função da velocidade e do ângulo de ataque, e a sua atitude (ângulos de arfagem, rolagem e guinada), sendo possível, portanto estimar também a trajetória.

A figura 2.1 ilustra algumas das grandezas que serão medidas (ângulo de ataque longitudinal - α , e lateral - β , e velocidade do avião com relação ao escoamento) e estimadas (sustentação, arrasto, sustentação lateral, momentos de arfagem, rolagem e guinada). A figura 2.2 ilustra um esboço do sistema de aquisição de dados a ser projetado.

Embora seja possível, a partir dos dados supostamente conhecidos (mencionados acima), estimar as forças e momentos aerodinâmicos, a trajetória, e a atitude da aeronave ao longo do tempo, essa estimativa é feita utilizando-se o modelo aerodinâmico estimado da aeronave (e as estimativas de perturbações atmosféricas). Desta forma, estes dados estimados ao longo do tempo não podem ser utilizados para controle autônomo, podendo ser utilizados somente para comparação com, e validação de resultados de outros experimentos, como testes em túnel de vento e vôos teste com outros sensores embarcados, como GPS e giroscópio. Caso o escopo desse trabalho fosse o controle autônomo (terceira etapa de projeto de um UAV), seria necessário embarcar GPS, bússola, giroscópio (ou DMU – *Dynamics Measurement Unit*) e altímetro.

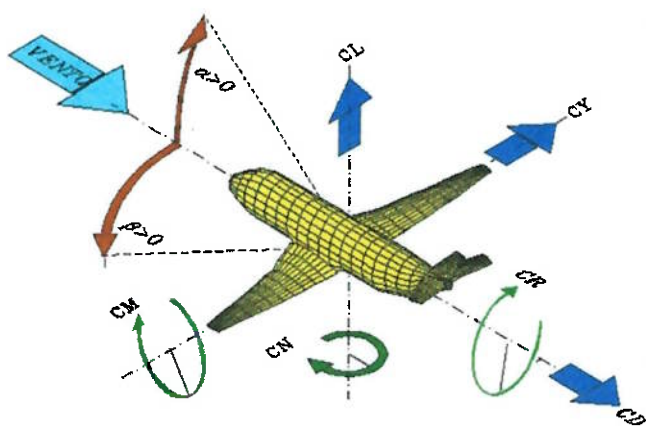


Figura 2.1 – Grandezas mensuradas e estimadas

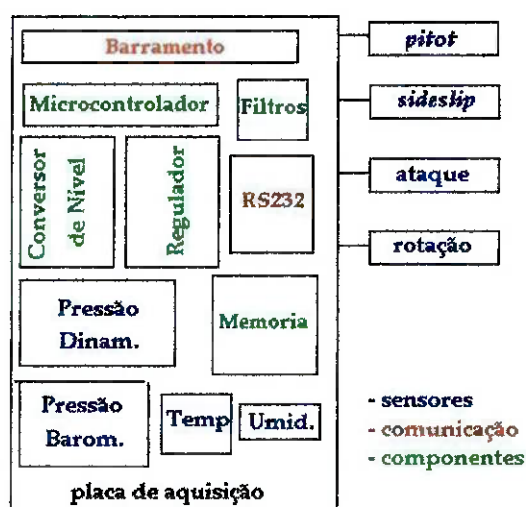


Figura 2.2 – Esboço da placa de aquisição

2.2. Projeto de Aquisição de Dados de 2004

Em 2004 as equipes Aclive e Poli Atlas participantes da *Poli aerodesign* projetaram e manufaturaram um sistema de aquisição de dados, com a finalidade de medir somente a velocidade da aeronave com relação ao escoamento de ar.

Foi utilizado neste projeto um PIC16F873, um HIN232CP (similar ao MAX232) para comunicação com o PC, e uma memória EEPROM serial (I2C) 24LC64. Os dados eram amostrados a 20 Hz e 10 bits (somente os 8 bits mais significativos eram armazenados, sendo possível portanto armazenar até 6,5 minutos de dados - um único voo). O sensor de pressão utilizado foi o modelo 850 da AutoTran Inc., que mede uma faixa nominal de ± 125 Pa, com saída de 1 a 5V, e é alimentado com 18V. Foram utilizadas duas baterias alcalinas Zn-Mn de 9V.

Antes do início do presente projeto, foram realizados, com o equipamento desenvolvido em 2004, dois testes em voo, e um teste em túnel de vento para calibração do sensor. Os resultados foram limitados (figura 2.4), e pode-se inferir que o sensor, ou algum componente (ou trilha) estava danificado(a), ou possivelmente o circuito estava sujeito a muito ruído eletromagnético. Testes mais aprofundados para solucionar este problema não foram realizados por questões de tempo.



Figura 2.3 – Projeto de 2004

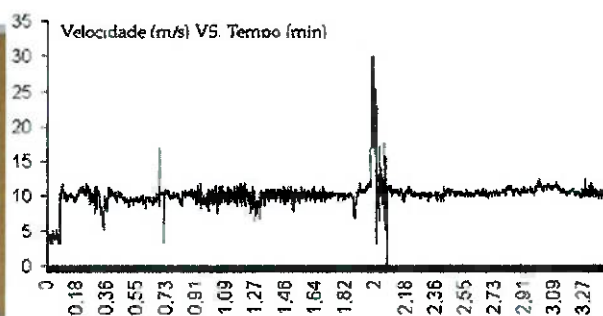


Figura 2.4 – Teste em voo, Velocidade por Tempo

2.3. Aeronaves Teste

As equipes da EPUSP participam do SAE Brasil Aerodesign desde 2000. As figuras 2.5 e 2.6 ilustram as aeronaves da equipe PoliActive TAM 2005, para as quais o sistema em questão foi desenvolvido. Os aeromodelos foram projetados e construídos por alunos de graduação da EPUSP, com orientação dos professores Antonio Luís Mariani e Alberto Hernandez Neto.

Pode ser visualizado pelas figuras 2.5 e 2.6 que a fuselagem possui um compartimento de carga. O sistema em questão será posicionado neste compartimento.

As tabelas 2.1, e 2.2 resumem, respectivamente, as principais características de desempenho, e geométricas das aeronaves.

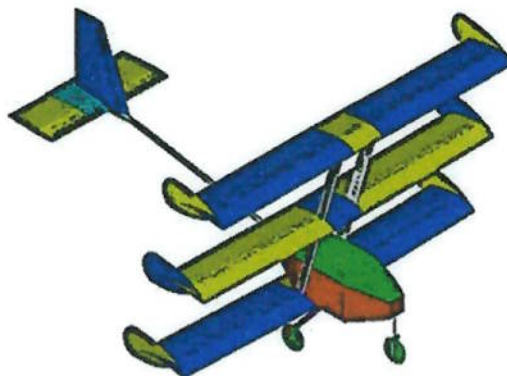


Figura 2.5 – Desenho em CAD da aeronave

Tabela 2.1 – Desempenho

Tempo max. de voo:	7 min.	Massa total (vazio):	4,5 kg
Alcance:	Limite visual	Carga útil máxima	8,855 kg
Velocidade decolagem:	12 m/s	Decolagem:	61 m
Velocidade máxima:	18 m/s	Pouso:	122 m

Tabela 2.2 – Características geométricas

Envergadura	1,82 m	Decalagem (inf. à sup.)	0°, 2°, 4°
Corda	0,24 m	Incidência (inf. à sup.)	0°, 2°, 4°
CG ao profundor	0,88 m	Razão de aspecto	7,85
CG à deriva	1,00 m	GAP	0,40 m
CG ao bico	0,30 m	Área alar	0,44 m ²
Enflexamento	0°	Área da deriva	0,04 m ²
Escalonamento	14°	Área do leme	0,03 m ²
Afilamento	1	Área do profundor	0,15 m ²
Diedro (asa inf. à sup.)	4°, 2°, 0°	Área total de aileron	0,05 m ²
Perfil da asa	Selig 1223	Perfil dos estabilizadores	Eppler168



Figura 2.6 – Foto da aeronave

Capítulo 3

Modelagem da Dinâmica de Aeronaves

Neste capítulo são apresentados conceitos e modelos matemáticos utilizados em sistemas de controle e navegação de aeronaves. Em futuras aplicações, estes conceitos serão utilizados, juntamente com o sistema de aquisição de dados.

Sistemas dinâmicos são caracterizados por equações diferenciais, lineares ou não, a parâmetros concentrados (EDOs) ou distribuídos (EDPs), variantes ou não no tempo. Na simulação da dinâmica de aeronaves pode-se trabalhar com diferentes tipos de equações diferenciais, dependendo da análise que se deseja fazer. Ao longo deste texto serão utilizadas EDOs e EDPs não variantes no tempo.

3.1. Referenciais e Nomenclatura

Na figura 3.1 definem-se os referenciais utilizados, e a nomenclatura das velocidades lineares (u , v , w) e angulares (p , q , r). São utilizados dois referenciais: um supostamente inercial fixo à Terra (X_f , Y_f , Z_f), e outro não inercial fixo à aeronave (x , y , z).

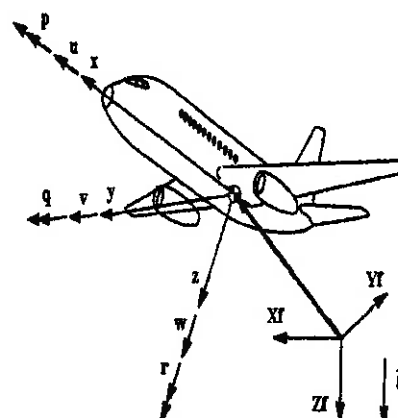


Figura 3.1 - Sistema de eixos ^[1]

A figura 3.2 define os movimentos de arfagem, rolagem e guinada de aeronaves, os momentos angulares (L , M , N), as taxas angulares (p , q , r), e as taxas lineares (u , v , w). O referencial (x , y , z) é aquele fixo a aeronave.

A figura 3.3 mostra de forma clara a atitude da aeronave com relação ao referencial fixo à Terra. Esses ângulos de atitude são chamados de ângulos de Euler e são utilizados, não somente em modelagem dinâmica de veículos aéreos, mas também na de veículos submersíveis, em robótica, etc. A nomenclatura para os diferenciais de profundor, leme, aileron, flap e tração mecânica são δe , δr , δa , δf e δt , respectivamente. Os ângulos de ataque e sideslip são dados, respectivamente, por α e β .

3.2. Modelagem

As leis físicas de cinemática e de dinâmica podem ser utilizadas para identificar as equações que regem o movimento de corpo rígido de aeronaves. Algumas hipóteses são feitas para obter essas equações gerais:

- Massa da aeronave constante no tempo;

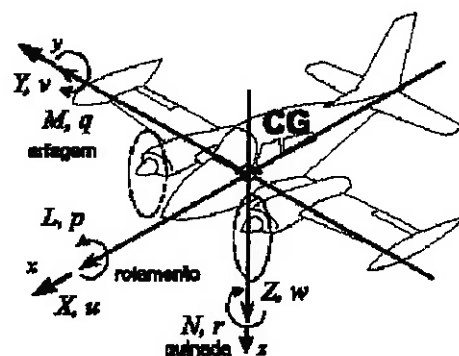


Figura 3.2 – Movimento da aeronave ^[6]

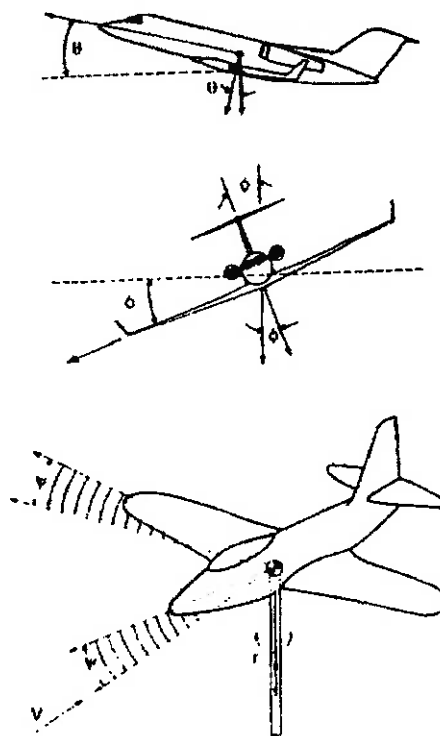


Figura 3.3 – Ângulos de Euler ^[1]

- Movimento não relativístico (baixa velocidade);
- Corpo perfeitamente rígido;
- Centro de massa fixo e coincidente com centro de gravidade;
- Momentos e produtos de inércia (relativos ao referencial da aeronave) constantes.

3.2.1. Modelo Dinâmico Não Linear

Tomando como premissas a segunda lei de Newton, e o teorema do movimento do baricentro, temos:

$$F = m \cdot \frac{d(v_c)}{dt} \quad (3.1)$$

A velocidade v_c é a velocidade do centro de massa da aeronave com relação ao referencial inercial (X_f, Y_f, Z_f). Agora supondo que deseja-se referenciar v_c , a um referencial não inercial, fixo ao centro de massa da aeronave (x, y, z) de tal forma que $v_c = (u \ v \ w)$, então a equação vetorial de equilíbrio de forças pode ser dada por:

$$F = m \cdot \frac{d(v_c)}{dt} + m \cdot (\omega \wedge v_c) \quad (3.2)$$

O termo $\omega = (p \ q \ r)$ é o vetor rotação do referencial não inercial (x, y, z), e v_c é dado neste mesmo referencial. Em uma situação genérica:

- A aeronave pode estar em uma atitude qualquer $(\theta \ \phi \ \psi)$;
- Forças aerodinâmicas mais tração motora são dadas por $(X \ Y \ Z)$ no referencial inercial;

Portanto, abreviando os termos *seno* e *coseno* por S e C , a equação 3.2 pode ser desenvolvida de modo a obter-se:

Equação geral de equilíbrio de forças (3.3)

$$\begin{aligned} X - m \cdot g \cdot S \theta &= m \cdot \left(\dot{u} + q \cdot w - r \cdot v \right) \\ Y + m \cdot g \cdot C \theta \cdot S \phi &= m \cdot \left(\dot{v} + r \cdot u - p \cdot w \right) \\ Z + m \cdot g \cdot C \theta \cdot C \phi &= m \cdot \left(\dot{w} + p \cdot v - q \cdot u \right) \end{aligned}$$

Analogamente, pode-se desenvolver as equações de equilíbrio de momentos, de modo que vetorialmente chega-se a:

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \frac{d(H)}{dt} + \omega \wedge H \quad (3.4)$$

$$\text{e ainda } H = \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} p & q & r \end{bmatrix} \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix} \quad (3.5)$$

O momento angular da aeronave (H), e os termos da matriz de inércia (I_x , I_y , I_z , I_{xy} , I_{xz} , I_{yz}), e o vetor rotação do referencial não inercial $[p \ q \ r]$, são todos dados com relação ao referencial não inercial fixo a aeronave.

Substituindo a equação 3.5 na equação 3.4 e desenvolvendo esta, chega-se a:

Equação geral de equilíbrio de momentos (3.6)

$$\begin{Bmatrix} L \\ M \\ N \end{Bmatrix} = \begin{bmatrix} I_x \cdot \dot{p} - I_{xz} \cdot \dot{r} + q \cdot r \cdot (I_z - I_y) - I_{xz} \cdot p \cdot q - \dot{q} \cdot I_{xy} \\ I_y \cdot \dot{q} - q \cdot r \cdot (I_x - I_z) + I_{xz} \cdot (p^2 - r^2) - \dot{p} \cdot I_{xy} - \dot{r} \cdot I_{yz} \\ I_z \cdot \dot{r} - I_{xz} \cdot \dot{p} + p \cdot q \cdot (I_y - I_x) + I_{xz} \cdot q \cdot r - \dot{q} \cdot I_{yz} \end{bmatrix}$$

A relação das variáveis cinemáticas, dadas no referencial não inercial, com as taxas lineares e angulares da aeronave, dadas no referencial inercial, é obtida através dos ângulos de Euler. Essas relações são obtidas em Nelson ^[1], e são mostradas a seguir:

Velocidade da aeronave dada no referencial inercial (3.7)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} C\theta.C\psi & S\phi.S\theta.C\psi - C\phi.S\psi & C\phi.S\theta.C\psi + S\phi.S\psi \\ C\theta.S\psi & S\phi.S\theta.S\psi + C\phi.C\psi & C\phi.S\theta.S\psi - S\phi.C\psi \\ -S\theta & S\phi.C\theta & C\phi.C\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Velocidade angular da aeronave dada no referencial inercial (3.8)

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & S\phi.\tan\theta & C\phi.\tan\theta \\ 0 & C\phi & -S\phi \\ 0 & S\phi.\sec\theta & C\phi.\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

3.2.2. Modelo Dinâmico Linear

As equações acima podem ser linearizadas usando a “Teoria das Pequenas Perturbações”.

As vantagens de se trabalhar com modelos lineares são:

- Fácil compreensão do fenômeno físico e da influência dos termos na dinâmica;
- Fácil e difundido tratamento analítico;
- Menor custo computacional;

As desvantagens de se trabalhar com um modelo linearizado são:

- Movimentos acoplados no modelo não linear não podem ser mensurados;
- Hipóteses de linearização limitam os casos em que o modelo pode ser usado.

A teoria das pequenas perturbações supõe que a aeronave esteja numa situação de equilíbrio, sendo todo e qualquer movimento uma pequena oscilação ao redor desta. Essa situação é válida especialmente quando a aeronave está em cruzeiro. Deste modo, as grandezas a serem analisadas podem ser dadas de acordo com a tabela 3.1.

Tabela 3.1 – Pequenas perturbações

$u = u_0 + \Delta u$	$X = X_0 + \Delta X$
$v = v_0 + \Delta v$	$Y = Y_0 + \Delta Y$
$w = w_0 + \Delta w$	$Z = Z_0 + \Delta Z$
$p = p_0 + \Delta p$	$L = L_0 + \Delta L$
$q = q_0 + \Delta q$	$M = M_0 + \Delta M$
$r = r_0 + \Delta r$	$N = N_0 + \Delta N$
$\delta = \delta_0 + \Delta \delta$	$v_0 = p_0 = q_0 = r_0 = \phi_0 = \psi_0 = 0$

Assume-se também que o diferencial das forças e dos momentos pode ser dado por uma série linear de Taylor em termos das velocidades e dos diferenciais das superfícies de controle, como mostrado na tabela 3.2.

Tabela 3.2 – Linearização do diferencial das forças e dos momentos

$\Delta X / m = X_u \Delta u + X_w \Delta w + X_\alpha \Delta \alpha + X_{\delta e} \Delta \delta e + X_{\delta f} \Delta \delta f + X_{\delta t} \Delta \delta t$
$\Delta Y / m = Y_v \Delta v + Y_p \Delta p + Y_\beta \Delta \beta + Y_r \Delta r + Y_{\delta r} \Delta \delta r$
$\Delta Z / m = Z_u \Delta u + Z_w \Delta w + Z_{w'} \Delta w' + Z_q \Delta q + Z_{\delta e} \Delta \delta e + Z_{\delta f} \Delta \delta f + X_{\delta t} \Delta \delta t + Z_\alpha \Delta \alpha$
$\Delta L / I_x = L_v \Delta v + L_p \Delta p + L_r \Delta r + L_\beta \Delta \beta + L_{\delta r} \Delta \delta r + L_{\delta a} \Delta \delta a$
$\Delta M / I_y = M_u \Delta u + M_w \Delta w + M_{w'} \Delta w' + M_q \Delta q + M_{\delta e} \Delta \delta e + M_{\delta f} \Delta \delta f + M_{\delta t} \Delta \delta t + M_\alpha \Delta \alpha$
$\Delta N / I_z = N_v \Delta v + N_p \Delta p + N_r \Delta r + N_\beta \Delta \beta + N_{\delta r} \Delta \delta r + N_{\delta a} \Delta \delta a$

Onde

$$X_u = \frac{\partial X}{\partial u} \cdot \frac{1}{m}, \quad L_v = \frac{\partial L}{\partial v} \cdot \frac{1}{I_x}, \quad M_u = \frac{\partial M}{\partial u} \cdot \frac{1}{I_y}, \quad N_v = \frac{\partial N}{\partial v} \cdot \frac{1}{I_z},$$

E assim por diante.

Considerando também que $\Delta\alpha \cong \Delta\theta$, e substituindo os termos das tabelas 3.1 e 3.2 nas equações 3.3 e 3.6, obtemos as equações linearizadas longitudinais e laterais:

Equações Longitudinais (3.9)

$$\begin{bmatrix} \dot{\Delta u} \\ \dot{\Delta w} \\ \dot{\Delta q} \\ \dot{\Delta \Theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & 0 & -g \\ Z_u & Z_w & V_{cruz} & 0 \\ M_u + M_w Z_u & M_w + M_w Z_w & M_q + M_w i_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \Theta \end{bmatrix} + \begin{bmatrix} X_{\delta} & X_{\delta T} \\ Z_{\delta} & Z_{\delta T} \\ M_{\delta} + M_w Z_{\delta} & M_{\delta T} + M_w Z_{\delta T} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta \delta T \end{bmatrix}$$

Equações Laterais (3.10)

$$\begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \\ \Delta \phi \end{bmatrix} = \begin{bmatrix} Y_{\beta}/V_{cruz} & Y_p/Cruz & (-1 + Y_r/V_{cruz}) & -g/V_{cruz} \\ L_{\beta} & L_p & L_r & 0 \\ N_{\beta} & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \\ \Delta \phi \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta r}/V_{cruz} \\ L_{\delta a} & L_{\delta r} \\ N_{\delta a} & N_{\delta r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta a \\ \Delta \delta r \end{bmatrix}$$

Capítulo 4

Estado da Arte

Este capítulo é responsável pela maior parte da revisão bibliográfica. Foram analisados vários projetos desenvolvidos no Brasil e em outros países.

O objetivo deste tópico é apresentar o contexto dos principais projetos mundiais de sistemas embarcados, e ilustrar as arquiteturas utilizadas. O enfoque é dado em sistemas embarcados em veículos autônomos.

Este capítulo está dividido em três partes. Na seção 4.1 são mostrados os principais projetos mundiais que envolvem sistemas embarcados visando ou controle autônomo, ou monitoração, ou controle remoto.

Na seção 4.2 são apresentados mais detalhadamente os seis principais projetos internacionais de UAVs e um nacional. O intuito nessa seção é o *benchmark*, focando em *hardware*. Os pontos importantes observados nessa seção são: missão, tipo de veículo adotado, arquitetura de hardware, e sensores utilizados.

No tópico 4.3 são apresentadas as principais empresas e instituições que produzem e comercializam aviãoica e/ou aeronaves autônomas, assim como os seus principais produtos.

4.1. Projetos de Sistemas Embarcados

Neste tópico são apresentados projetos de veículos autônomos e/ou sistemas embarcados, com foco na missão e no tipo de veículo adotado.

4.1.1. ARMAR



Figura 4.1 – ARMAR ^[30]

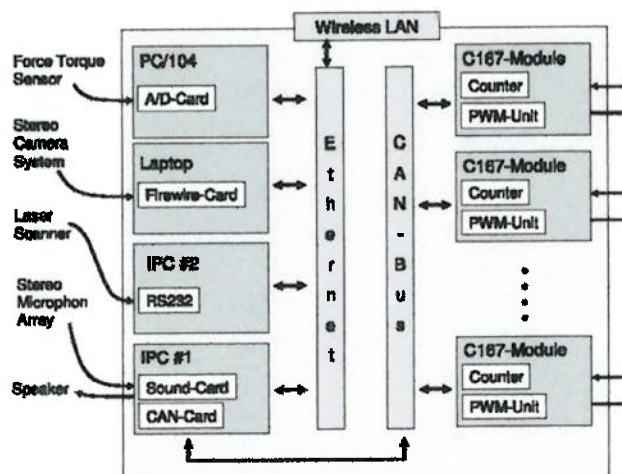


Figura 4.2 – Arquitetura embarcada no ARMAR ^[30]

Este robô ^[30] foi desenvolvido no “*Research Center for Information Technologies*”, Alemanha. O objetivo deste projeto é obter um humanoide capaz de interagir naturalmente com situações pré-definidas, e que seja capaz de realizar tarefas que necessitem de mobilidade.

Foram utilizados dois PCs (1 GHz cada) para navegação, cálculo de trajetória e controle da fala. Para os algoritmos da visão foi utilizado um computador *laptop* (1.7 GHz). O controle dos motores dos braços foi efetuado utilizando-se um PC/104 (700 MHz). A comunicação entre os módulos se dá através de barramentos CAN e via rede Ethernet (baseado em TCP/IP). O sistema operacional utilizado foi o Linux, kernel 2.4.20 com interface de tempo real RTAI 2.4.1.11.

4.1.2. OLGA

Este projeto ^[17] foi desenvolvido no Departamento de Engenharia Elétrica e Ciências da Computação da Universidade da Califórnia, EUA, e seu objetivo é obter um helicóptero autônomo não tripulado.



Figura 4.3 – OLGA ^[17]

A figura 4.4 mostra a arquitetura do sistema de controle OLGA. Atualmente esse sistema de controle é comercializado em Zürich – Suíça como “*wePilot1000*”.

No OLGA, o computador de bordo roda a 200 MHz. De acordo com o autor, o maior esforço computacional se dá na fusão sensorial, cujo objetivo é a navegação. O sistema operacional adotado é o RT HelyOS.

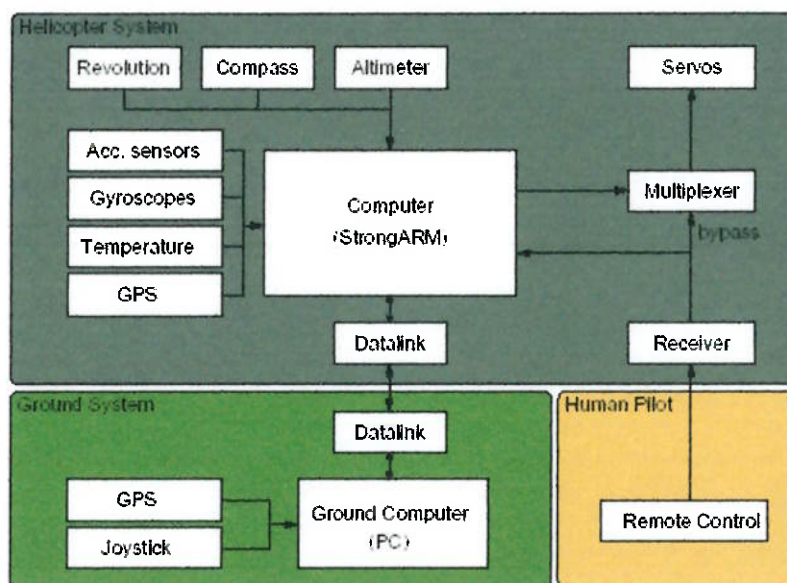


Figura 4.4 – Arquitetura do sistema de controle OLGA ^[17]

4.1.3. GenBase II

Este robô ^[28] foi desenvolvido na “*Faculty of Technology Technical Computer Science, University of Bielefeld*”, Alemanha. Seu objetivo é obter alta precisão de posicionamento, com alto desempenho, em uma unidade autônoma. Para se atingir tal objetivo o autor utilizou filtros de Kalman estendidos no controle e uma arquitetura de controle robusto.

A figura 4.6 mostra o sistema embarcado no GenBase II. O sistema de controle foi baseado em PC, com sistema operacional Linux. A comunicação entre os módulos se dá via serial e via rede *wireless* (TCP/IP).



Figura 4.5 – GenBase II ^[28]

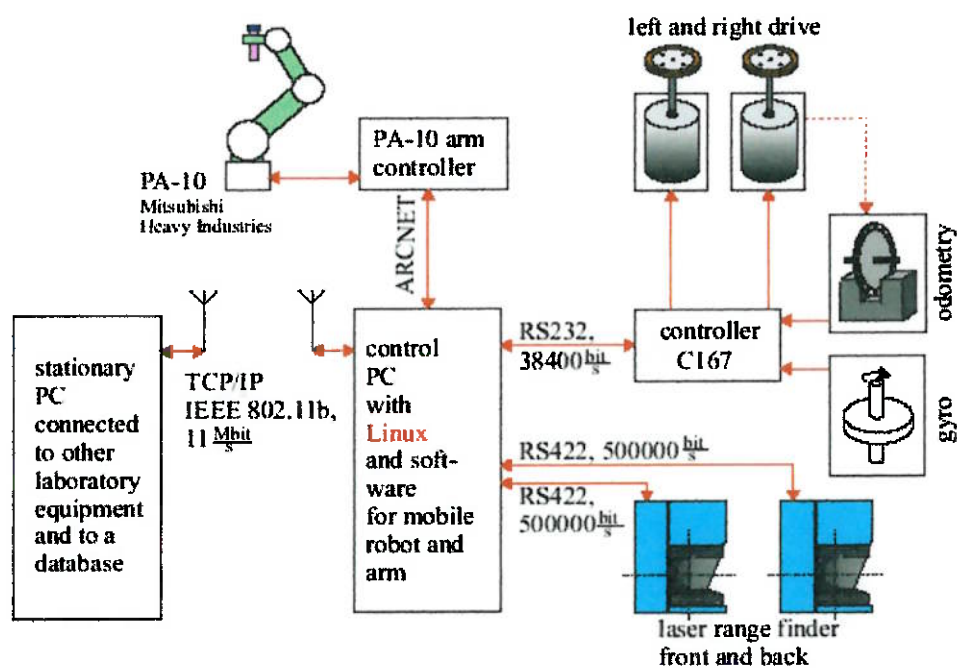


Figura 4.6 – Arquitetura embarcada no GenBase II ^[28]

4.1.4. Sistema de Controle para Múltiplos Veículos Autônomos

Este projeto ^[16] também foi desenvolvido no Departamento de Engenharia Elétrica e Ciências da Computação da Universidade da Califórnia, EUA. Seu objetivo é implementar um sistema de controle flexível, para controle de múltiplos veículos autônomos. Este trabalho é, na verdade, uma expansão do OLGA, apresentado em 4.1.2, porém foram realizadas mudanças de *hardware* e total reformulação do *software* de controle.

Neste projeto sistema operacional possui *kernel* de tempo real. O sistema de controle utiliza fusão sensorial para navegação.

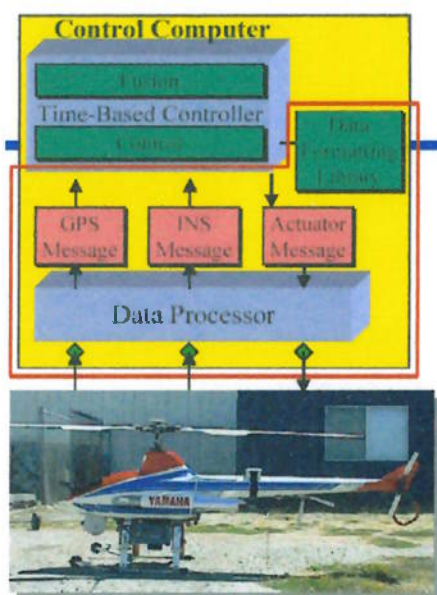


Figura 4.7 – Hardware e plataforma ^[16]

4.1.5. Navigator

Este projeto ^[25] foi desenvolvido no Virginia Tech, EUA. O objetivo é projetar uma arquitetura que possibilite controle autônomo e remoto.



Figura 4.8 – Navigator ^[25]

Este projeto utiliza um PC com dois Pentium III 450MHz. A arquitetura adotada é modular, e a comunicação entre os módulos e o computador central é feita através de rede Ethernet. A aquisição dos sinais amostrados é gerenciada por um PLC.

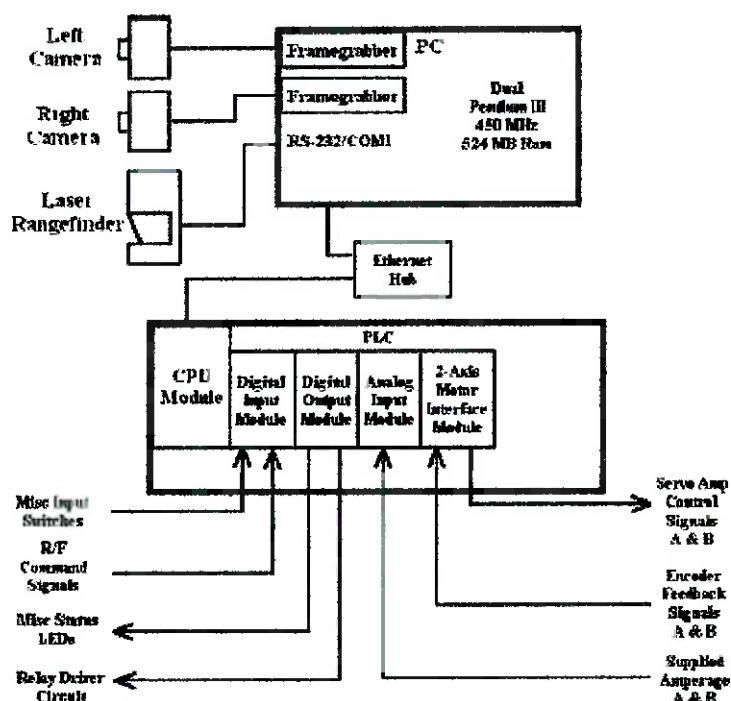


Figura 4.9 - Sistema embarcado no Navigator ^[25]

4.1.6. Sistema de Reconhecimento Aéreo Autônomo

Este projeto ^[15] foi realizado na Universidade do Arizona, EUA. Foi desenvolvido um sistema autônomo de controle e navegação para múltiplas aeronaves.



Figura 4.10 – Aeronaves teste e PAD ^[15]

O intuito é programar uma missão que será executada autonomamente por todas as aeronaves, mesmo que a missão seja diferente para cada aeronave.

A figura 4.10 ilustra as aeronaves e o PAD (que simula um míssil). Os sensores utilizados são giroscópios, GPS, sensores de pressão dinâmica e barométrica, de temperatura, e sensores de impacto. A figura 4.11 mostra o esquema do sistema embarcado e base.

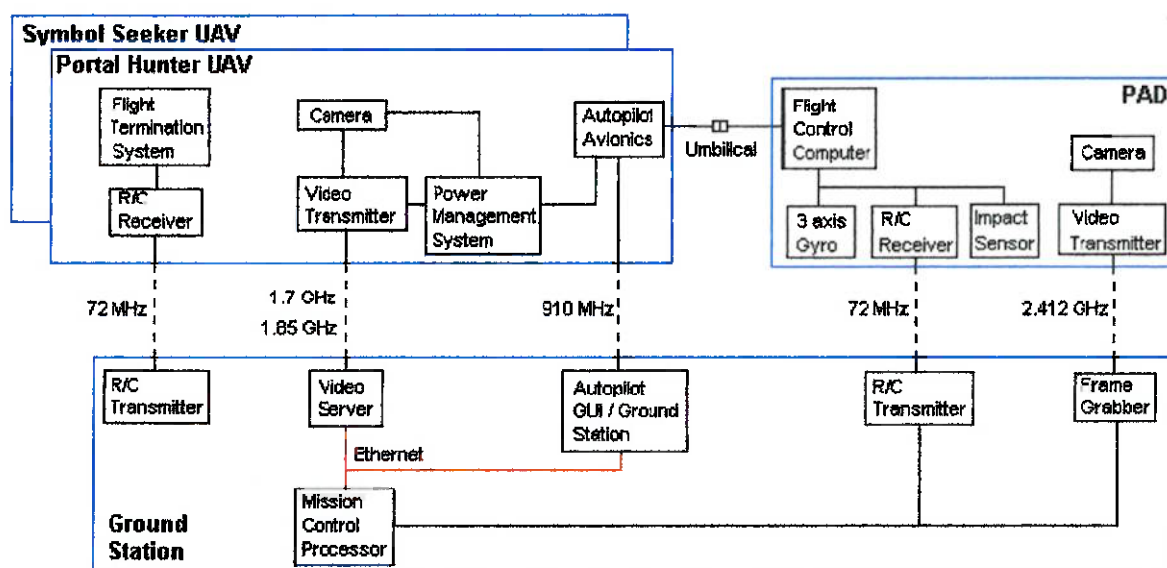


Figura 4.11 – Sistemas embarcado nos UAVs, nos PADs, e da estação em solo ^[15]

4.1.7. Plataforma Experimental e Controle de Helicóptero

Este trabalho ^[20] foi realizado na Universidade de Brasília, e tem como objetivo a concepção de uma plataforma teste e o controle de helicópteros, visando, em estudos futuros, obter um veículo autônomo. A figura 4.12 mostra o aparato final. A figura 4.13 ilustra o esquema da eletrônica adotada. Como o autor utilizou motores elétricos no rotor principal e no de cauda, foi necessário um circuito de acionamento.



Figura 4.12 – Plataforma experimental ^[20]

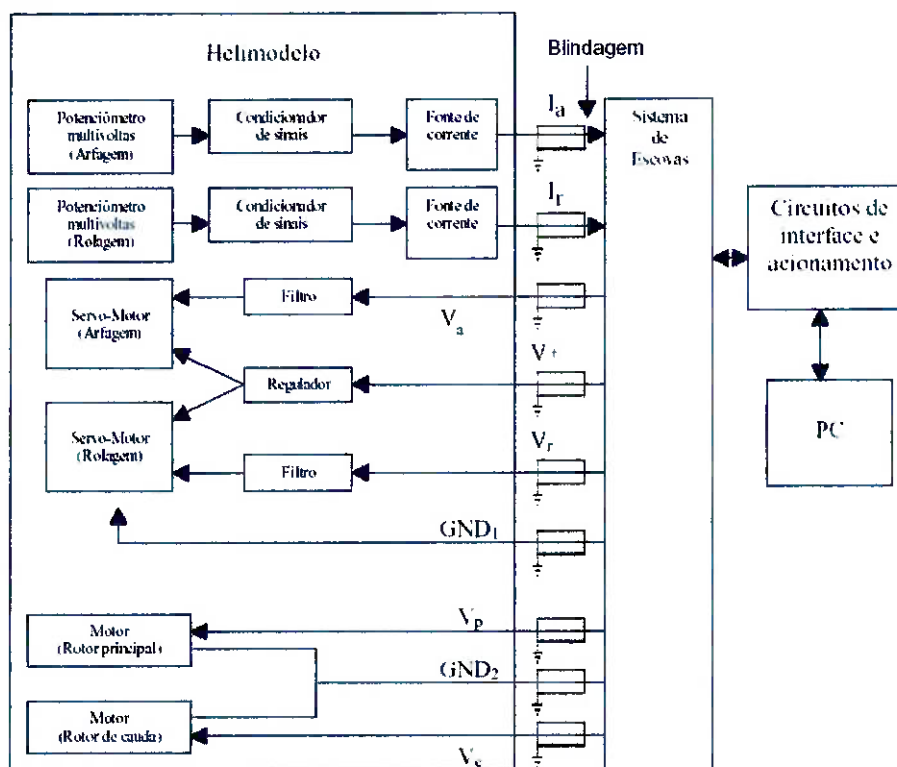


Figura 4.13 – Diagrama de blocos do circuito eletrônico ^[20]

4.1.8. Tethra

Este projeto ^[22] foi desenvolvido na Universidade de Limerick, Irlanda. Trata-se de um veículo submersível autônomo. A figura 4.15 ilustra o esquema da eletrônica.



Figura 4.14 – Tethra ^[22]

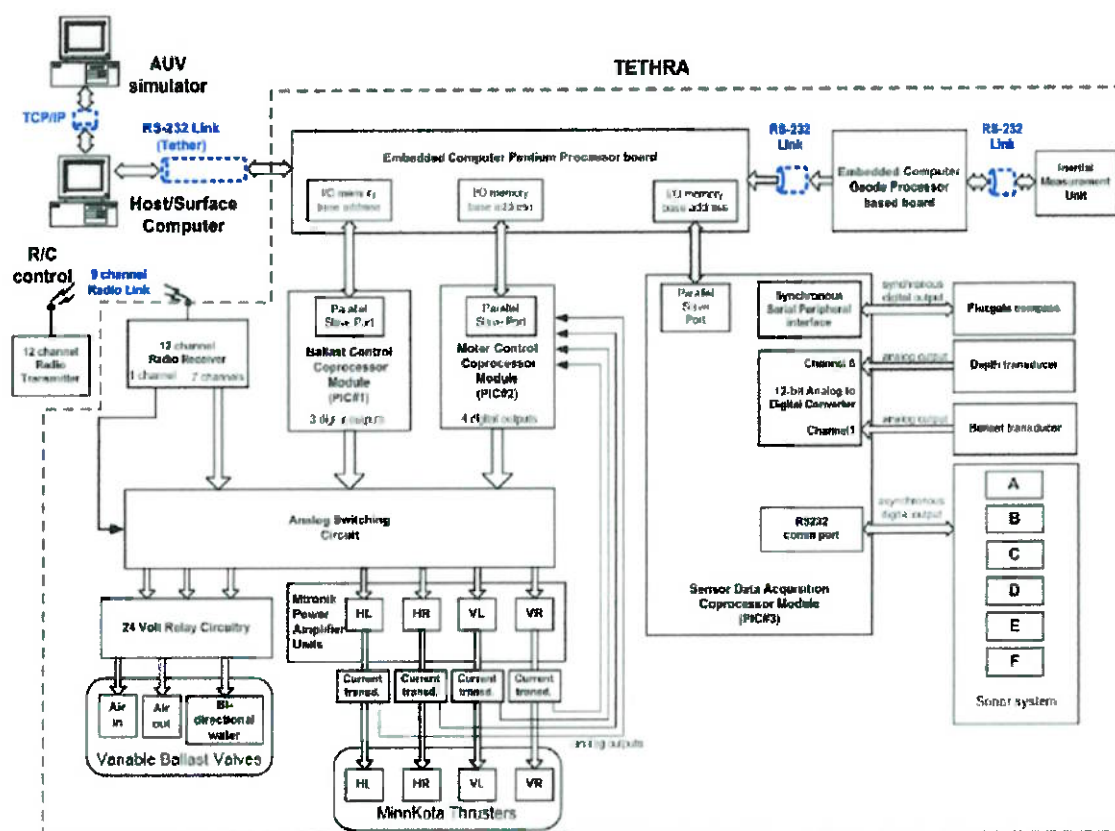


Figura 4.15 – Sistema Eletrônico do Tethra ^[22]

Neste projeto foi utilizada uma arquitetura mista: computador mais microcontroladores. Para aquisição de dados e controle dos motores foram utilizados microcontroladores, e para armazenamento e cálculos foram utilizados microcomputadores. Os módulos se comunicam via serial. Observa-se também um recurso comum, que é a multiplexação dos sinais analógicos.

4.1.9. *Smart Car*

Este projeto ^[24] foi desenvolvido na Universidade Estadual de Nova York, EUA. O objetivo é obter um veículo de baixo custo, robusto e autônomo, que execute tarefas em tempo real, e que possa ser tele-operado por múltiplos usuários, também em tempo real.

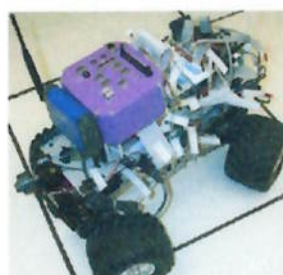


Figura 4.16 – *Smart Car* ^[24]

Foram utilizados um PC/104 para controle e navegação, um módulo para leitura e armazenamento de dados de cada sensor, e um módulo para controle dos motores. A transferência de dados entre a base e a plataforma é realizada via rede Ethernet.

4.2. Projetos de Aeronaves Autônomas

Neste tópico são detalhados seis principais projetos internacionais de UAVs, e um nacional.

Tem-se também como objetivo dessa seção o *benchmark*, de forma que se possa aprender boas práticas de projeto, e propor soluções que evoluam das dificuldades enfrentadas pelos autores das referências citadas.

4.2.1. *Dragon Fly*

O projeto *Dragon Fly* ^[7] foi desenvolvido no Departamento de Aeronáutica e Astronáutica da Universidade de Stanford, EUA.



Figura 4.17 – *DragonFly* ^[7]

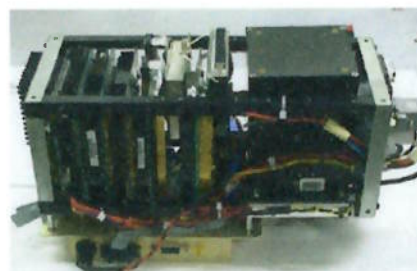


Figura 4.18 – Eletrônica embarcada ^[7]

A equipe embarcou no modelo: 4 antenas DGPS; sensor de velocidade; sensor de ângulo de ataque longitudinal e lateral; UMI (unidade de medição inercial); PC/104 600 MHz; modem e receptor. O sistema foi isolado mecanicamente e eletricamente, adotando uma carcaça metálica, conforme figura 4.18. Os testes tinham como objetivo a identificação da dinâmica linear e não linear do modelo, assim como verificação de desempenho do controle autônomo e remoto. O sistema operacional adotado foi o QNX Neutrino. Foi utilizado um sistema de controle robusto de tempo real. O tratamento de dados foi feito no espaço da frequência.

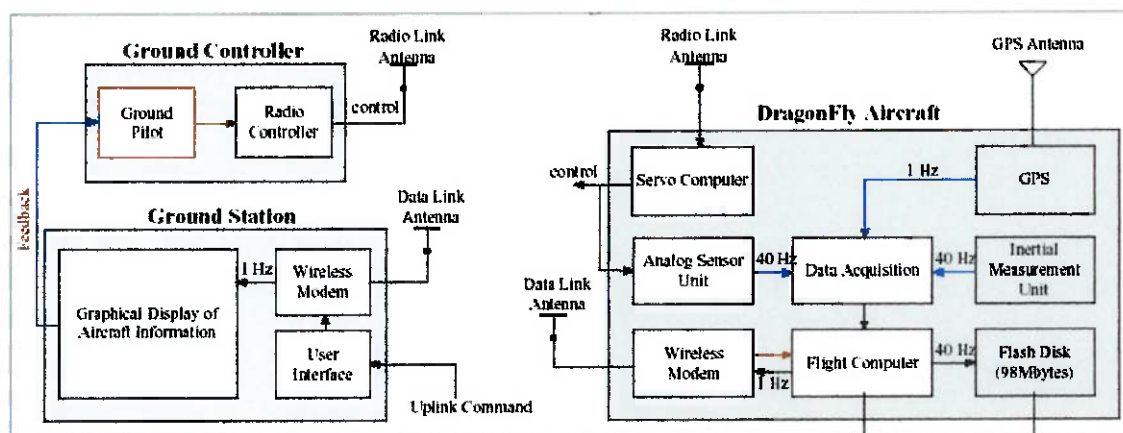


Figura 4.19 – Esquema eletrônico do *Dragon Fly* ^[7]

4.2.2. Projeto Piloto de UAV

Este projeto piloto ^[9] foi desenvolvido no Departamento de Sistemas Elétricos e Automação da Universidade de Pisa, Itália, e seu foco é o desenvolvimento do *software*.

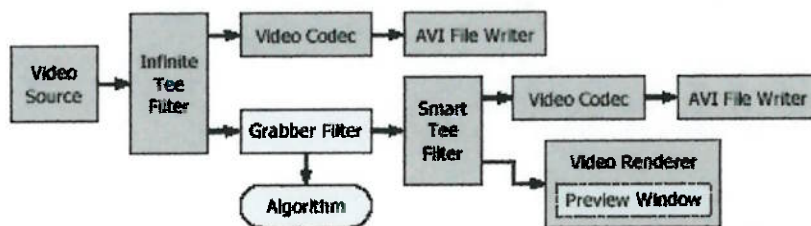


Figura 4.20 – Algoritmo para tratamento do sinal de vídeo ^[9]



Figura 4.21 – Plataforma de testes ^[9]

O *hardware* é composto de receptor Futaba com oito canais, PC/104 (256 Mb RAM, 266 MHz, HD de estado sólido de 128 Mb rodando QNX), bateria Ni-MH 2Ah, receptor DGPS, UMI com seis graus de liberdade, e câmera de vídeo. O objetivo é obter um sistema estável, confiável, de baixo custo, flexível, e tele-operado em tempo real.

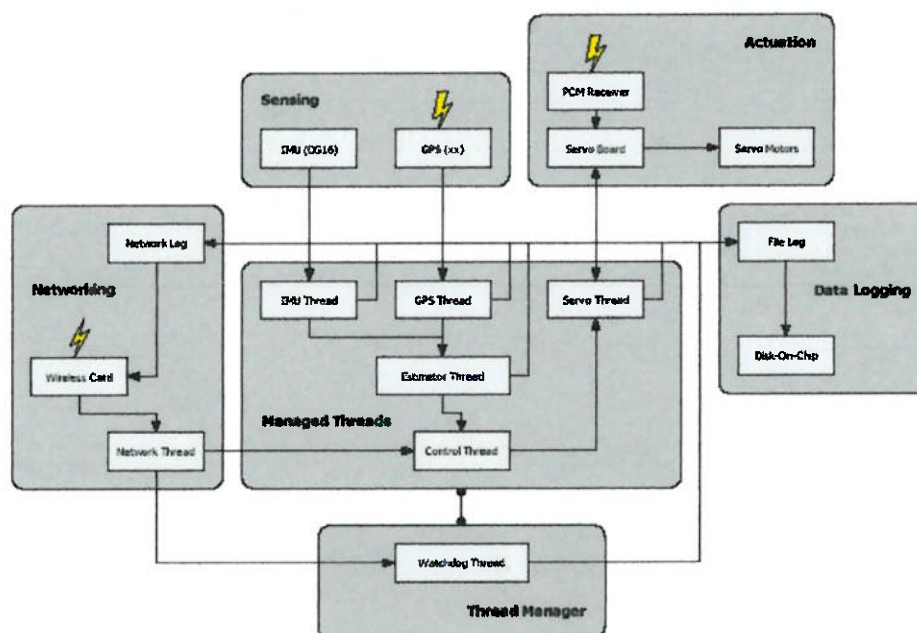


Figura 4.22 – Arquitetura de *software* ^[9]

4.2.3. Mini UAVs

O projeto de mini UAVs ^[10] foi desenvolvido na Universidade Brigham Young, EUA, em parceria com a força aérea americana. O objetivo é testar o piloto automático, o sistema de navegação, e o sistema de monitoração remota.

Foram embarcados: micro câmera, receptor e antena DGPS, sensor de pressão absoluta, sensor de pressão relativa, três giroscópios, três acelerômetros, um transmissor de vídeo, um *transciever*, e um receptor de RC.



Figura 4.23 – Plataformas de testes ^[10]

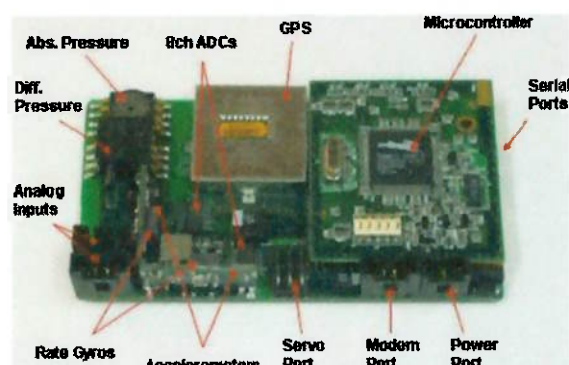


Figura 4.24 – Eletrônica embarcada ^[10] (60,4 g)

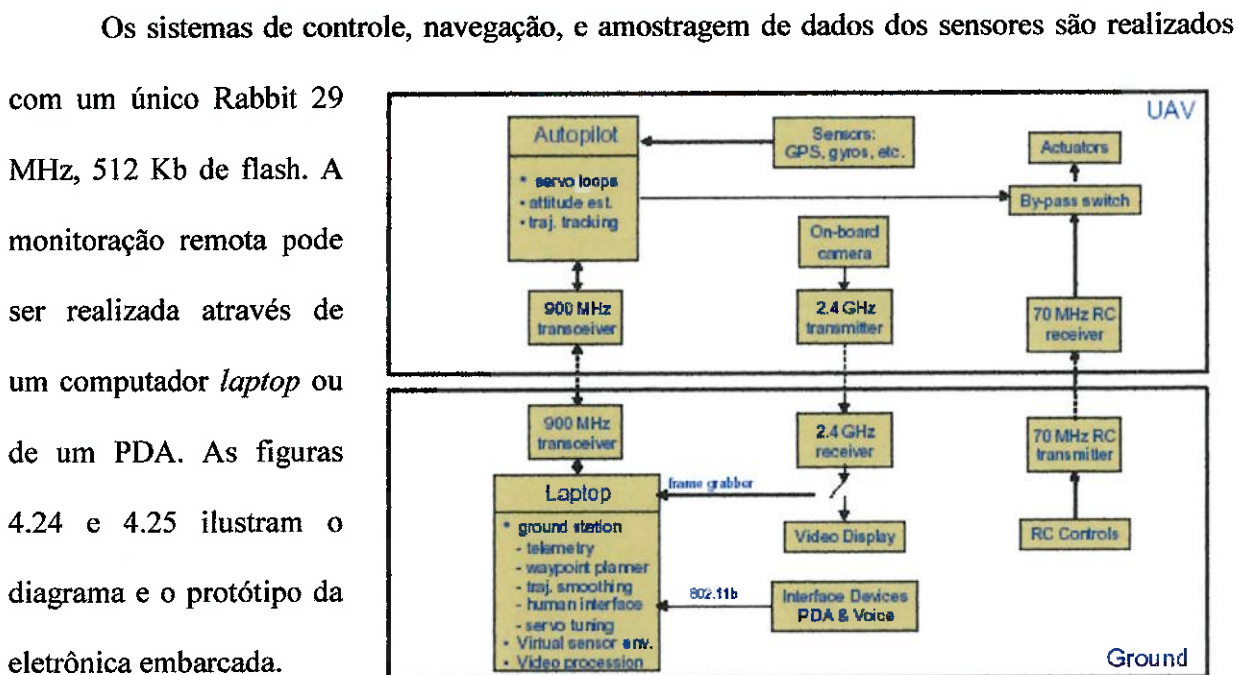


Figura 4.25 – Diagramas de bloco da eletrônica ^[10]

4.2.4. Sistema de Controle Adaptativo para UAVs

Este projeto ^[12] foi desenvolvido no Instituto de Tecnologia de Geórgia, EUA, com o objetivo de obter um sistema de baixos peso e consumo, e alto desempenho. Foram utilizados FPGA e DSP.



Figura 4.26 – UAV Gtmax ^[12]



Figura 4.27 – Avionica ^[12]

O circuito do FPGA realiza a leitura dos sensores e o controle dos atuadores (figura 4.28), assim como armazena os dados em memória. O DSP realiza o tratamento dos dados e os cálculos de controle e navegação.

Foram embarcados:

- Receptores RC e DGPS;
- Três “rate gyros”;
- Três acelerômetros;
- Sensor de altitude;
- Sensor velocidade;
- Sensor temperatura;
- Câmera de vídeo;
- Transmissor de vídeo;

O sistema operacional adotado no DSP foi o RT-MicroC OS II.

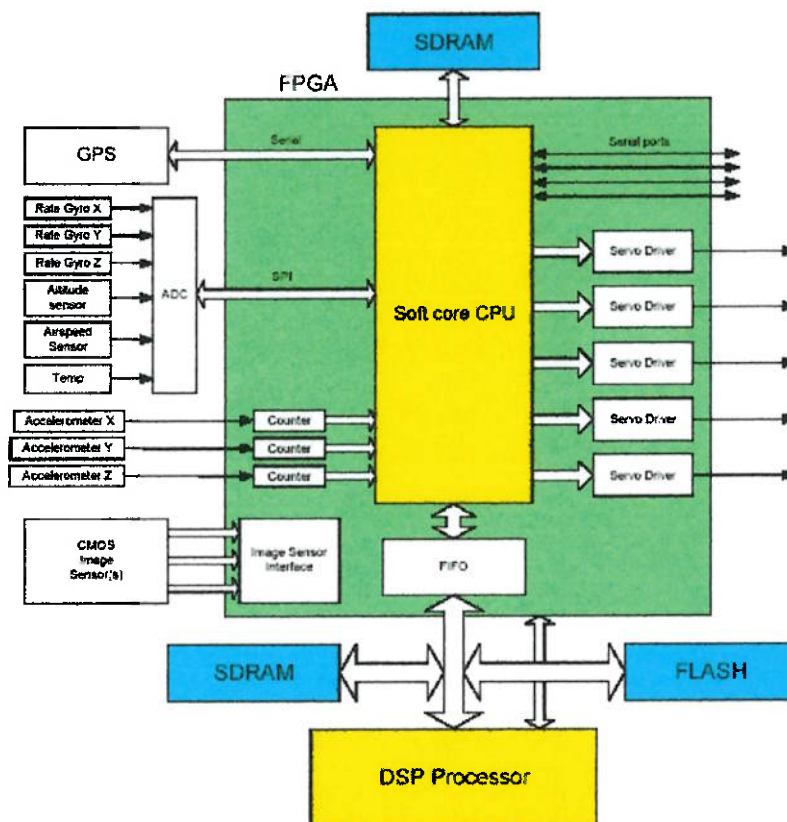


Figura 4.28 – Diagrama de blocos do sistema ^[12]

O autor também menciona como contornou problemas de vibrações, ruídos, resfriamento, etc.

4.2.5. Piper J3

Este projeto ^[8] foi desenvolvido no Departamento de Computação e Ciência da Informação da Universidade de Pensilvânia, EUA. O objetivo é obter um sistema expansível, flexível, que possibilite o controle e a navegação autônomos em tempo real de múltiplos UAVs, assim como a tele-operação deles.



Figura 4.29 – Piper J3 ^[8]

Para tanto, foram utilizados:

- Receptor DGPS e UMI;
- Câmera e controle desta;
- Sensores de pressão;
- Receptor RC e rede *wireless*;
- Transmissor de vídeo.

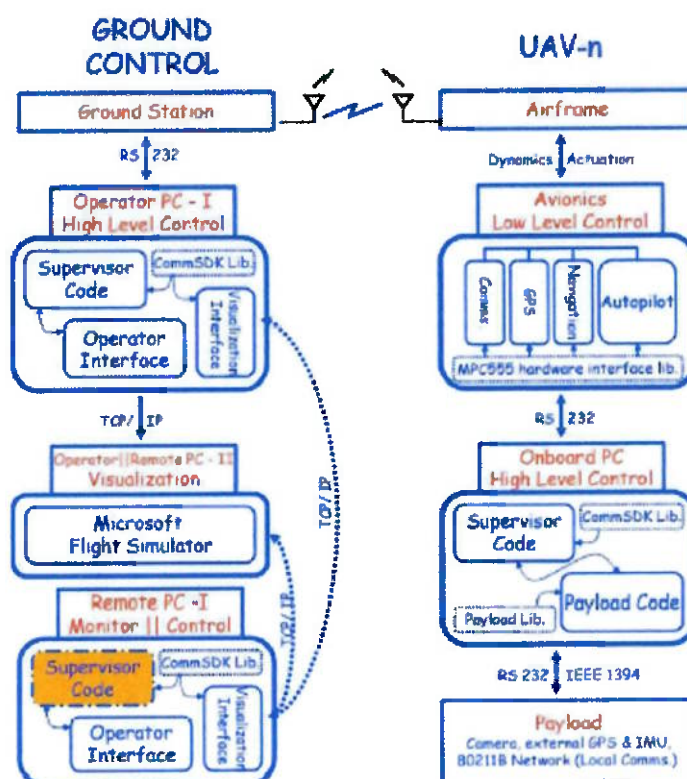


Figura 4.30 – Diagrama de blocos do sistema ^[8]

Em cada aeronave foi embarcado um computador *laptop* Dell X200. Foi desenvolvida também uma estação base, utilizando-se de um PC, como ilustra a figura 4.30. Neste projeto, o sistema embarcado é caracterizado por dois módulos, um microcontrolado responsável pelos acionamentos, e outro baseado em PC, responsável pelo controle da câmera, e pela telemetria.

4.2.6. Mini Child



Figura 4.31 – *Mini Child* ^[47]



Figura 4.32 – Avionica ^[47]

Este projeto ^[47] foi desenvolvido no Instituto de Tecnologia de Massachusetts (MIT), EUA. O objetivo é implementar um sistema de controle ótimo robusto em duas aeronaves, utilizando arquitetura embarcada diferente em ambos os casos.

Foram utilizados: UMI, receptor DGPS, sensor de pressão dinâmica, sensor de pressão barométrica, circuito microcontrolado de condicionamento de sinal, receptores FUTABA, transceiver, duas vídeo câmeras, transmissor de vídeo, e um PC104.

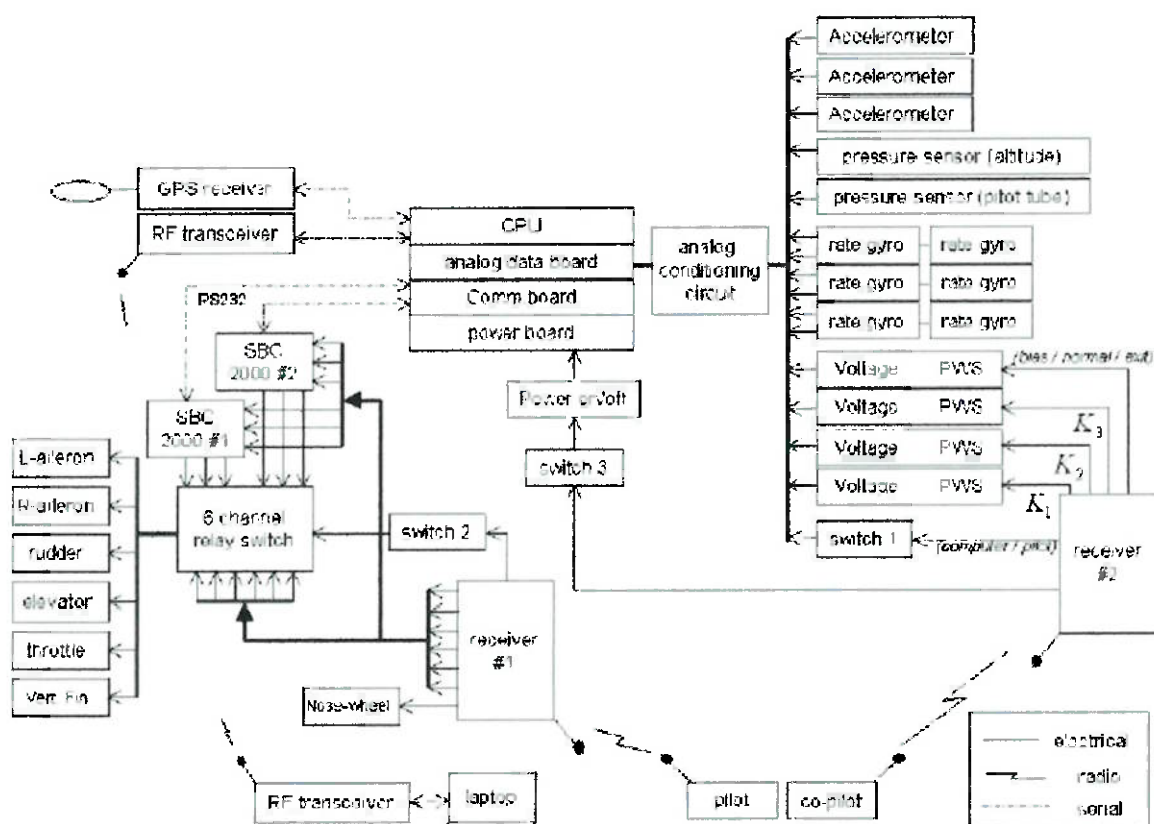


Figura 4.33 – Eletrônica embarcada no *Mini Child* ^[47]

4.2.7. ARARA

Este projeto ^[44] ^[48] ^[49] é uma parceria entre USP, Embrapa e a AGX Tecnologia, e foi desenvolvido na Universidade de São Paulo, campus de São Carlos. Trata-se do maior projeto acadêmico de UAVs do Brasil.

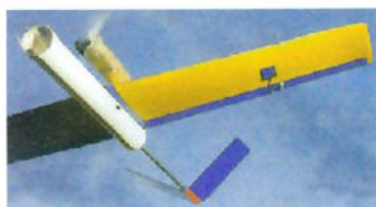


Figura 4.34 – ARARA

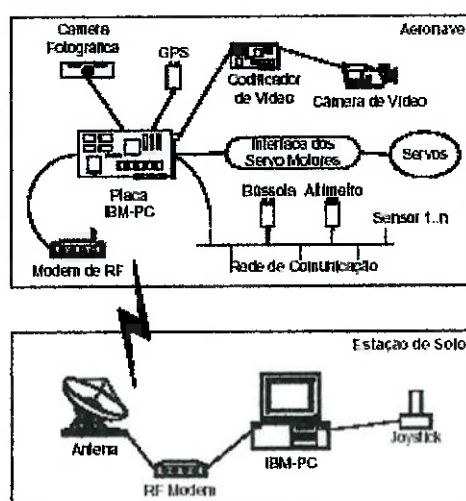


Figura 4.35 – Hardware ^[44]

Esse projeto foi dividido em 4 etapas. A fase I representa o projeto e construção do aeromodelo rádio-controlado, com uma câmera abordo, acionada também pelo rádio. A fase II representa a implementação do sistema embarcado com transmissão de dados via RF. Porém a aeronave ainda é rádio-controlada. Na fase III é implementado o sistema de controle autônomo. A fase quatro é implementação de missões com base no sensoriamento e no tratamento das imagens. Atualmente o projeto encontra-se na fase IV.

A figura 4.35 mostra um esquema do *hardware* utilizado. Foram embarcados: receptor DGPS, altímetro, bússola, UMI, transmissor de vídeo, modem RF, câmeras, e um PC/104.

4.3. Empresas e Produtos Comercializados

A tabela 4.1 relaciona as empresas e seus principais produtos relacionados à UAVs e aviação. No Brasil, o CTA e está atualmente, em parceria com a Embravant e a Avibrás, desenvolvendo UAVs, e já em 2006 testará as versões preliminares destes projetos.

Tabela 4.1 – Principais empresas de UAVs e aviação

Empresa	País	Produto
Prince Airmodel	Brasil	Aeromodelo com aviação embarcada
AGX Tecnologia	Brasil	Aeromodelo com aviação embarcada
Mectron	Brasil	UAV, aviação
Avibrás	Brasil	UAV, aviação
Embravant	Brasil	UAV
Global Aerial Surveillance	EUA	UAV, aviação
Elbit Systems	EUA	UAV, aviação
BAE Systems	EUA	UAV, aviação
Honeywell	EUA	UAV, Aviação
Boeing	EUA	UAV, aviação
ATK	EUA	UAV, aviação
UAV Flight Systems Inc.	EUA	UAV, aviação
DRS Tech.	EUA	UAV, aviação
Northrop Grumman	EUA	UAV, aviação
Aurora	EUA	UAV, aviação
Kaman	EUA	Aviação
Dara Aviation	EUA	UAV

Capítulo 5

Projeto e Especificações do *Hardware* Embarcado

O objetivo deste capítulo é mostrar o projeto e especificações do *hardware* embarcado. O projeto segue a metodologia ilustrada na figura 5.1, definida pelo Prof. Dr. Jun Okamoto Jr ^[50].

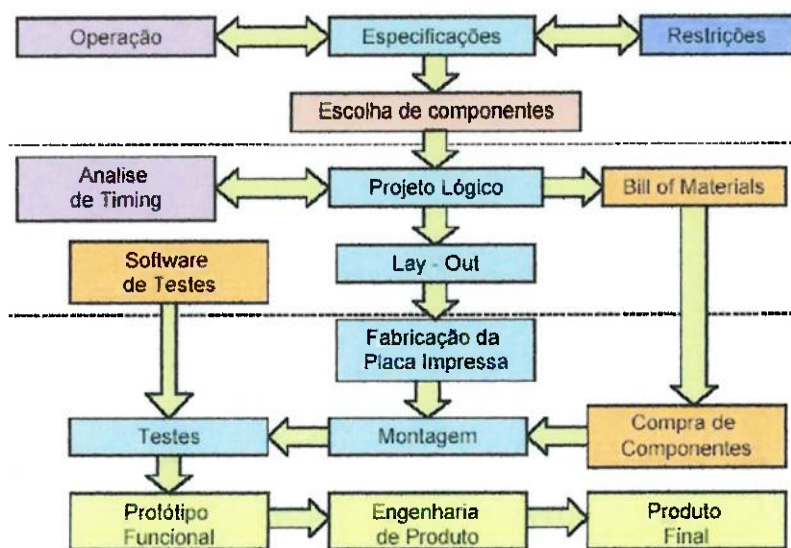


Figura 5.1 – Metodologia adotada no projeto do *hardware* ^[50]

5.1. Requisitos e Especificações

Os requisitos analisados são: massa (kg), dimensões (m), período de amostragem (s), capacidade de armazenamento de dados (kbytes), capacidade de comunicação externa, e

capacidade de processamento, consumo de energia elétrica, imunidade a ruído, expansibilidade, e custo. A seguir mostra-se a análise de cada um dos requisitos.

- **Requisito de Massa:**

Este quesito é analisado considerando o apresentado no capítulo 2, tal que as aeronaves teste foram projetadas para carregar 8,855 kg no nível do mar. É preciso também lembrar que o sistema projetado deve ser expansível visando a terceira etapa de um projeto de uma aeronave autônoma, ou seja, outras placas serão conectadas a essa placa de aquisição, e que a aeronave deve voar a cerca de 1500 m de altitude-densidade. Desta forma a placa deste projeto, junto com sensores e baterias deve apresentar no máximo 2 kg. Esta premissa supõe que em futuras aplicações serão embarcadas mais três placas com aproximadamente a mesma massa.

- **Requisito de Consumo de Energia Elétrica:**

As aeronaves teste têm capacidade máxima de sete minutos de voo. Desta forma, somando o tempo de ligar e desligar o sistema, uma bateria deve ser capaz de durar no mínimo dez minutos.

Visando diminuir custos, o ideal seria utilizar uma bateria alcalina comum não recarregável de nove volts, de Zn – Mn. Supõe-se que a aeronave possa voar até seis vezes em um dia, sem recarregar a bateria, somando-se uma hora. A corrente drenada pelo circuito é estimada em 1,5 A, além disso, a tensão de alimentação deve ser maior que 7,0 V (entrada mínima no regulador). Essas informações sugerem que a bateria tenha vida mínima até 1500mAh, com tensão mínima de 7,0 V. A pilha não recarregável com maior tensão inicial e menor inclinação da curva de descarga é a pilha Li – MnO₂, cuja curva é mostrada na figura 5.2.

Pelo gráfico verifica-se que uma pilha de Li – MnO₂ não satisfaz os requisitos (nem pilhas Zn-Mn), sendo necessário a utilização de oito dessas pilhas, por dia de vôo! Opta-se, portanto, por uma bateria, ou um *pack* recarregável, de Ni-MH ou Li-Ion (baterias Ni-Cd com corrente drenada acima de 1A sofrem queda de tensão abrupta antes de uma hora de uso, portanto não são adequadas). Essas baterias e *packs* têm um alto custo, variando de US\$10,00 a US\$400,00. A figura 5.3 ilustra a curva de descarga de uma pilha Ni-MH modelo HR-3U da Sanyo. Neste caso seriam necessárias sete dessas pilhas (AA) no *pack*, com duração de 200 ciclos.

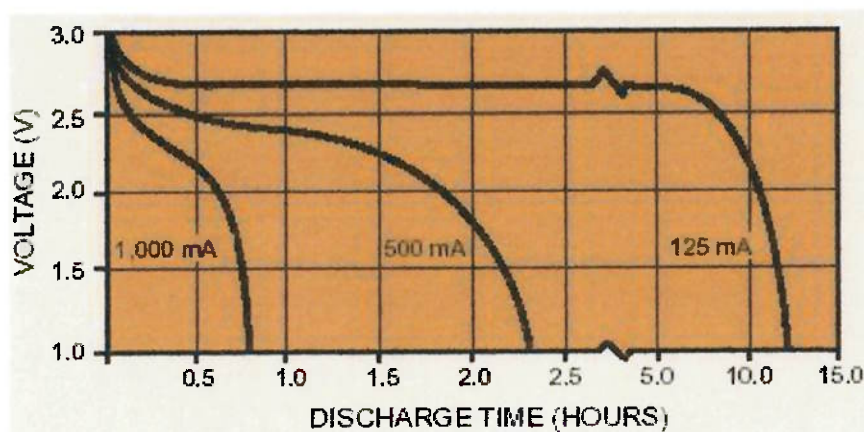


Figura 5.2 – Curva de descarga, bateria Li-MnO₂

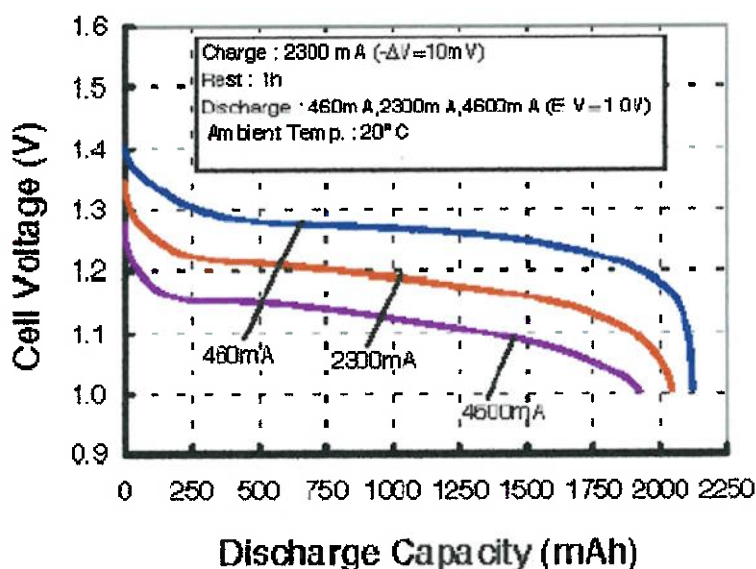


Figura 5.3 – Curva de descarga, pilha Ni-MH

- Requisito de Dimensões:

Devido ao limite de volume do compartimento de carga, a placa terá um limite de 11 cm por 14 cm, por 6 cm de altura, supondo que no futuro serão empilhadas outras três placas.

- Requisito de Período de Amostragem:

A resposta dinâmica mais rápida da aeronave é estimada em 20 Hz. Desta forma, a menor frequência possível para futuro controle da aeronave, dado pelo Teorema da Amostragem (para reconstrução de sinais), é 40 Hz. Adota-se então um mínimo de 50 Hz. Desta forma todos os sensores devem ser capazes de amostrar a no mínimo 50 Hz. De fato, em termos de controle, a frequência melhor seria $10 \times 20 \text{ Hz} = 200 \text{ Hz}$, porém a princípio adotemos 50 Hz.

- Requisito de Capacidade de Armazenamento de Dados:

Para vôos com no máximo dez minutos, seis vôos em um dia, armazenando sete variáveis mensuradas (pressão barométrica, pressão dinâmica, temperatura, umidade relativa, ângulo de ataque, ângulo de *sideslip*, e rotação da hélice), com uma avaliação inicial de dez bits cada, a 50 Hz cada, temos $10 \text{ bits/sensor} \times 7 \text{ sensores} \times 50 \text{ Hz} \times 6 \text{ vôos} \times 10 \text{ min/vôo} = 1,51 \text{ Mbytes}$.

- Requisito de Comunicação Externa:

Neste projeto, todos os dados serão lidos *offline*, com aeronave em terra. Os dados serão transmitidos para um PC e tratados nele. Desta forma, a maneira mais simples e barata para comunicação com PC é através de uma porta serial RS232. Outro detalhe importante é a possível futura expansão deste sistema, de modo a se acoplar a outros módulos. É necessário, portanto, um

barramento para comunicação, onde serão disponibilizados oito bits de dados, mais a comunicação serial, sinais de alimentação, terra, e controle do barramento.

- Imunidade ao Ruído:

O baixo nível de tensão e corrente em sistemas embarcados (desse porte), permite que esses sistemas sofram muita interferência eletromagnética, podendo ocorrer a perda dos sinais amostrados e até mesmo a falha do circuito. Para minimizar o máximo possível, os fios dos sensores deverão ser blindados, e o circuito deverá ser enclausurado em uma armadura metálica.

- Requisito de Capacidade de Processamento:

Superestima-se uma sequência de quinhentos ciclos de *clock* do microcontrolador, para cada sinal amostrado. Logo o *clock* do microcontrolador deve ter uma frequência mínima de $50 \text{ Hz/sensor} \times 7 \text{ sensores} \times 500 \text{ ciclos de clock} = 0,2 \text{ MHz}$. Adota-se 1 MHz à princípio.

5.2. Escolha de Componentes

Este tópico detalha a escolha dos sensores, circuitos integrados (CIs), microcontrolador, e memória.

Atualmente existe uma variedade muito grande de nível de tensão (lógica) que os CIs operam (e.g. TTL, CMOS, ETL). Grande parte dos CIs CMOS são compatíveis com CIs TTL, outra parte necessita de *buffer* CMOS para compatibilizar. No Brasil, os CIs mais comuns são TTL, e em menor escala CMOS, por isso vamos adotar somente CIs TTL e CMOS neste projeto.

A figura 5.4 ilustra o nível de tensão das principais lógicas.

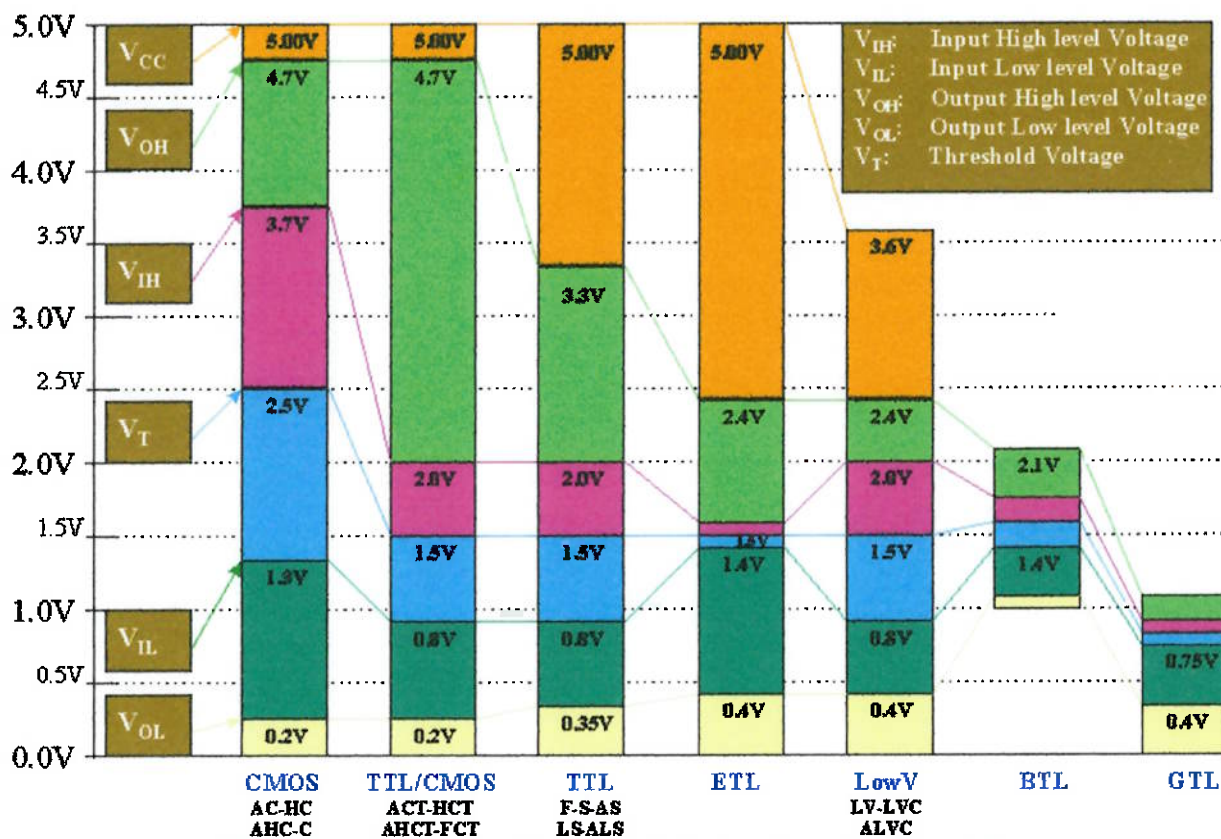


Figura 5.4 – Faixa de tensão dos níveis lógicos dos principais padrões

5.2.1. Sensores

Em primeira instância adota-se uma faixa de operação da saída dos sensores de 0 à 5V (devido ao fato de conversores A/D com lógica TTL usualmente trabalharem nessa faixa), com 10bits de conversão A/D, desta forma, obtendo uma resolução de 5 mV. Deve-se lembrar também dos requisitos de amostragem, consumo, peso, e dimensões.

A Equipe Poli Active TAM recebeu como apoio para o desenvolvimento deste trabalho dois sensores de pressão, um microcontrolador, e um sensor de atitude de três eixos da empresa Freescale do Brasil.

As especificações de todos os sensores selecionados constam no anexo A.

- Sensores de Pressão:

A pesquisa pelos sensores de pressão é resumida na tabela 5.1. Verifica-se que, embora os sensores doados pela Freescale não sejam os mais apontados para essa aplicação (por apresentarem uma faixa ampla de pressão), um deles será adotado nesse projeto devido ao alto custo da melhor opção: DC001NGC4 da Honeywell (US\$125). Esse sensor da Honeywell é dito a melhor opção, pois além de estar exatamente na faixa de pressão de trabalho (200Pa, equivalente a velocidades de 0 à 20 m/s), ele apresenta uma alta sensibilidade.

Tabela 5.1 – Sensores de pressão								
Product Family	Pressure Rating Maximum (kPa)	Over Pressure (kPa)	Supply (Typ)(Vdc)	Sensitivity (mV/kPa)	Accuracy (0°C to 85°C) (% of VFSS)	Pressure Type	\$	OBS.:
MPXV5004	4.0	10	5.0	1000	±1.5	D, V	9.64	Freescale
DC001NGC4	0.250	2.5	5	8000	±2.0	G,D	125	Honeywell
MPXH6101	102	400	4.6	54	±1.8	A	9.36	Freescale
MPX4100	105	400	4.6	54	±1.8	A	9.36	Freescale

O sensor Freescale MPXV5004DP – pressão diferencial – tem uma saída de 1,0 a 1,16 V (para velocidades de 0 a 18m/s), com desvio de $\pm 0,07$ V no valor medido, e uma sensibilidade de 1 mV/Pa. Desta forma, para fins de controle, este sensor é inaceitável, mas para teste deste projeto, o mesmo é utilizado, já que não representa nenhum custo. Esse sensor é alimentado com 5 V e drena 10 mA.

- Sensor de Temperatura:

A tabela 5.2 mostra os principais sensores aplicáveis. Todos os modelos tabelados têm aproximadamente as mesmas características elétricas, variando pouco de uma para outro a sensibilidade e a tensão de saída.

Tabela 5.2 – Sensores de temperatura

Product Family	Temp. range (°C)	Acuracy (°C)	Sensibility (mV/K)	Output to 25°C (typ) (V)	Supply Voltage (V)	\$	OBS.:
LM135A	-55 - +150	1	10	2.98	5	11.9	National
LM235A	-40 - +125	1	10	2.98	5	7.50	National
LM335Z	-40 - +100	1	10	2.98	5	2.00	National
TC1047A	-40 - +125	1	10	0.8	5	0.19	Microchip
MCP9701	-40 - +125	1	19.5	1	5	0.19	Microchip
MAX6605	-55 - +125	0.75	11.9	1.1	5	1.80	Maxim
DS600	-40 - +125	0.5	6.45	0.8	5	1.80	Maxim
F50	-40 - +125	2	10	0.75	5	0.457	Fairchild

Devido à disponibilidade no mercado nacional, e às dimensões, vamos adotar o sensor LM335Z da National Semiconductor. Este sensor é na verdade um diodo zener, com tensão zener variável de acordo com a temperatura.

O LM335Z pode ser utilizado em diversas configurações, aumentando ou diminuindo seu desempenho. O circuito com este componente pode ser alimentado com 5V, drena uma corrente máxima de 10 mA, a saída ocupa a faixa de 2,0 V à 3,5 V, com um erro máximo de 1°C na faixa de -40°C a 100°C, e sensibilidade de 10 mV/K. A figura 5.7 ilustra o LM335Z.

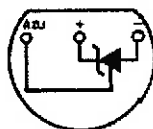


Figura 5.7 – Representação do sensor LM335

- Medidor de Rotação:

Para medição de rotação de um eixo, são usualmente utilizados tacômetros digitais (*encoders*) acoplados diretamente ao eixo do motor. No caso desta aplicação, é impossível acoplar um *encoder* ao eixo do motor, dado que só existe uma extremidade aberta, justamente onde a hélice é posicionada. Existem outras possibilidades utilizando, por exemplo, sensores:

- Capacitivos, fixando um adesivo metálico na hélice;
- Indutivos, fixando um adesivo metálico na hélice;
- Foto sensíveis, fixando um adesivo refletor na hélice.

Estudando essas outras três possibilidades, verifica-se que as opções com sensores capacitivos e indutivos são descartadas, pois a frequência máxima, dos melhores modelos, não ultrapassa 5000 rpm (a rotação máxima do motor com hélice chega a 12000 rpm), além de serem alimentados com mais de 10 V.

A solução evidente então é utilizar sensores foto sensíveis ao espectro infravermelho (*IR*) (medem até $5 \cdot 10^5$ rpm). O conceito básico é implementar um circuito que emita um pulso a cada rotação da hélice, utilizando um *LED IR*, um foto diodo *IR*, e um adesivo refletor na hélice.

A maior dificuldade em trabalhar com este tipo de configuração é a interferência eletromagnética que o foto diodo pode sofrer com a radiação ambiente. Para evitar este problema será utilizado um circuito oscilador com uma frequência conhecida, de forma que o sinal recebido pelo foto diodo seja contabilizado com um pulso somente se ele estiver na frequência conhecida.

Ao selecionar um *LED IR* deve-se verificar: a potência dissipada, a alimentação, a queda de tensão e corrente máxima, a máxima frequência de chaveamento, a curva de intensidade luminosa por abertura angular, a curva de intensidade luminosa por comprimento de onda, e a faixa de operação de temperatura. A potência irradiada indica a máxima distância que o sinal será percebido com determinada intensidade. A frequência de chaveamento é utilizada para enviar

- Utilizando potenciômetros (ou capacitores variáveis) acoplados a eixos presos a perfis aerodinâmicos livres;

Analisando a primeira opção, verifica-se que esta apresenta custo e massa relativamente elevados. A segunda opção requer três sensores de pressão, e a segunda somente dois potenciômetros (ou capacitores variáveis). Desta forma, a solução mais viável economicamente e mais simples é a terceira. A idéia básica com essa solução é acoplar um potenciômetro ao eixo de um perfil aerodinâmico livre (o eixo pode girar livremente - salvo o torque resistivo do potenciômetro - sob ação do escoamento). Para o perfil rotacionar sobre ação do escoamento é necessário determinar as dimensões do perfil, para que o torque resistivo do potenciômetro seja vencido. Deve-se utilizar um perfil aerodinâmico que se estabilize na direção do escoamento, ou seja, um perfil simétrico, com uma região que, no caso de escoamento, garanta uma maior distribuição e concentração de pressão no bordo de fuga, com relação ao bordo de ataque. Desta forma o perfil tenderá a estabilidade na direção do escoamento. A diferença entre o ângulo de incidência da asa e o ângulo de estabilização do perfil livre é o ângulo de ataque da asa.

5.2.2. Circuitos Integrados

Os circuitos integrados usualmente utilizados em projetos de eletrônica são:

- Amplificador Operacional e portas lógicas (utilizados em filtros, lógica digital, etc);
- *Bus Transceiver* (controle *tri-state* do barramento);
- Reguladores (5V para nível TTL e 3,3V para CMOS);
- Conversor de nível (TTL – RS232), conversores A/D e D/A;
- *Buffers* (suprimento de corrente para alimentação de várias entradas lógicas);
- Multiplexadores (mux) e demultiplexadores (demux);
- Chaves digitais e analógicas;
- Contadores;
- Circuito somador;
- Circuito comparador;
- Circuitos osciladores;
- Registradores e *latches*;

Esses CIs e circuitos listados acima são comumente empregados na Eletrônica, e para aplicações de baixa potência, as famílias TTL e CMOS são as mais utilizadas.

Por ser de conhecimento geral, a descrição dos CIs adotados ao longo do projeto é mostrada somente no anexo B.

Esse tipo de memória possui um controlador de memória *flash* e também uma interface IDE (*Integrated Device Electronics*) semelhante à usada em HDs. Geralmente, os cartões *Compact Flash* usam memória *flash* tipo NAND.

- *Smart Media* (SM):

O cartão de memória *Smart Media* (cujo nome original é *Solid State Floppy Disk Card*) chama a atenção logo de início por ser parecido com os já defasados disquetes de 3.5".

Lançado pela Toshiba em meados de 1995, esse tipo de cartão tem tamanho de 45 mm por 37 mm por 0,76 mm e sua fabricação é simples se comparada com as tecnologias concorrentes. Essa característica deve-se ao fato desse cartão não possuir circuitos controladores, reduzindo seu custo. Devido a isso, as funções de controle tiveram que ser inseridas nos equipamentos que trabalham com *Smart Media*. Isso representa uma desvantagem na questão do aumento de capacidade, pois estes aparelhos "não sabem" trabalhar com cartões que armazenam mais dados do que o programado em seus circuitos. Como consequência, os cartões *Smart Media* mais comuns possuem tamanho de até 128 MB e estão cada vez mais em desuso.

- *Multi Media Card* (MMC):

Lançado em 1997 através de uma parceria entre a SanDisk e a Siemens, o cartão de memória *Multi Media Card* é um dos mais pequenos existentes (24 mm por 32 mm por 1,4 mm) e mesmo assim é capaz de armazenar até 256 MB.

Esse tipo de cartão foi desenvolvido inicialmente para telefones celulares e *paggers*, mas só se tornou conhecido e, conseqüentemente, utilizado em câmeras digitais, MP3 *Players* e outros dispositivos, depois que a *PalmOne* o inseriu em seus *hand helds* (computadores de mão). O tipo

de *flash* geralmente utilizado no MMC é a NAND. Um outro modelo do MMC é o RS-MMC (*Reduced Size Multimedia Card*).

- *Secure Digital* (SD):

Lançado no final de 2001, o cartão *Secure Digital* é derivado do MMC. Seu desenvolvimento se deve ao fato da indústria detentora de direitos autorais reclamar do MMC por este permitir livremente a troca de qualquer tipo de arquivo, o que para a indústria fonográfica, por exemplo, representa risco de pirataria.

Diante disso, a Toshiba trabalhou numa nova versão do MMC e essa tecnologia recebeu recursos contra cópias de materiais protegidos por direitos autorais e também um barramento de transferência de dados mais rápido que o do MMC (em cerca de 4 vezes). Além disso, os cartões SD contam com uma pequena trava que impede a gravação ou eliminação acidental de informações.

Um cartão SD também possui tamanho reduzido (24 mm por 32 mm por 2,1 mm) e, armazenam até 1 GB. Existem dispositivos que trabalham com SD que também são compatíveis com MMC. O tipo de *flash* utilizado no SD é o NAND.

- *Memory Stick*:

O cartão *Memory Stick* foi desenvolvido pela Sony e é usado principalmente em câmeras fotográficas e filmadoras digitais. Embora seja menor que o *Compact Flash* e o *Smart Media*, seu tamanho não é dos mais reduzidos: 50 mm por 21,5 mm por 2,8 mm.

O *Memory Stick* também conta com proteção contra gravação ou eliminação acidental de arquivos (apenas em uma versão conhecida como *MagicGate*). Ainda, ele pode ser utilizado em

versão ROM (*Read-Only Memory*), ou seja, seu conteúdo pode apenas ser lido, mas não apagado (como acontece com os CDs). Esse recurso é interessante para a distribuição de softwares.

- *eXtreme Digital* (xD):

O cartão de memória xD é um dos mais recentes. Foi disponibilizado em 2002 pela Fujifilm e Olympus (sendo usado apenas em câmeras digitais destas empresas) e sua fabricação geralmente é feita pela Toshiba (o que deixa seu preço elevado). Teoricamente, a capacidade dos cartões xD pode chegar a até 8 GB. Seu tamanho é o menor encontrado até agora: 20 mm por 25 mm por 1,7 mm.

Visto como um substituto do *SmartMedia*, o xD também não tem circuitos controladores internos, o que significa que aparelhos mais antigos não serão capazes de ler cartões do tipo com nova capacidade de armazenamento. Por não ser muito popular, as empresas envolvidas neste tipo de cartão tiveram até que desenvolver adaptadores para que o xD seja lido em interfaces *PC-Card* (antiga PCMCIA) e *Compact Flash*.

A fim de selecionar o tipo de cartão utilizado, foram considerados o custo, a disponibilidade no mercado nacional, as dimensões físicas, e a simplicidade na implementação. O MMC foi adotado por possuir taxa de transferência de até 2,5 Mbytes/s, possuir custo relativamente menor, e ser um dos menores fisicamente. Os cartões MMC trabalham com lógica CMOS, portanto, será implementado um divisor resistivo para compatibilizar o nível *HIGH* de TTL para CMOS. A descrição detalhada do MMC encontra-se no anexo C.

são interessantes nesse projeto, pois este deve ser reprogramado em aplicações futuras. Deseja-se também (por questões de montagem) que o encapsulamento seja do tipo PDIP (figura 5.9).

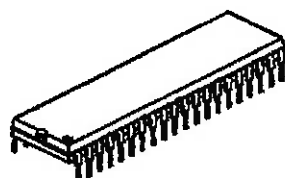


Figura 5.9 – Microcontrolador com encapsulamento PDIP de 40 pinos

O MCU em questão pode ser de 8 bits, pois não será utilizado para nenhum cálculo. Como não há necessidade de processamento pesado (sem cálculos, sem análise de sinais, pois isso será feito *offline*) então também não há necessidade de se utilizar um DSP (*Digital Signal Processor*). Com base nessas afirmações, foi realizada uma pesquisa e constatou-se que os modelos PIC16F877 e PIC16F874 da Microchip se enquadram perfeitamente nos requisitos. Outro fabricante de renome Atmel, também possui um modelo que se enquadra: AT89C51AC3. A Freescale-Motorola também tem um modelo que se enquadra: MCS08GT60. Como apoiadora deste trabalho, a Freescale doou não somente dois sensores de pressão e um inclinômetro de três eixos (citados anteriormente), mas também doou uma unidade deste MCU. A tabela 5.3 resume as características dos modelos mencionados.

Tabela 5.3 – Microcontroladores

Model	ADC	ADC (bits)	USART/SPI	Supply (V)	Prog. Mem. (kbytes)	RAM (bytes)	OBS.:
PIC16F874A	8	10	1 / 1	5	flash – 7	192	Microchip
PIC16F877A	8	10	1 / 1	5	flash – 14,3	368	Microchip
AT89C51AC3	8	10	1 / 1	5	flash – 64	2304	Atmel
MCS08GT60	8	10	2 / 1	3,3	flash – 60	4096	Freescale

Todos esses modelos trabalham com frequência de *clock* igual ou superior a 20MHz, podem ser reprogramados, usam encapsulamento PDIP 40 ou 42, e estão na faixa de preço de US\$4,50 a US\$5,00. O modelo da Freescale opera segundo a lógica CMOS (alimentação de

3,3V, conforme tabela 5.3). Embora, as quatro soluções sejam possíveis, decide-se usar o PIC16F877A, pois este é mais usual no mercado nacional (sendo, portanto, de mais fácil reposição e obtenção de material didático para a sua programação). Este MCU opera até 20MHz, existe no encapsulamento PDIP 40, é re-programável via ICSP, é largamente utilizado em projetos eletrônicos, é de fácil reposição, e satisfaz todos os requisitos. A descrição detalhada desse MCU encontra-se no anexo D.

5.3. Projeto Esquemático

Todo o projeto do circuito (e do *layout*) foi realizado no software *Eagle* versão 4.13 *shareware* (www.cadsoft.de/freeware.htm). As seguintes considerações são feitas:

- Na placa serão posicionados os sensores de temperatura, umidade, pressão barométrica e diferencial. O circuito do medidor de rotação também será posicionado na placa, apenas o LED IR e o foto diodo IR será posicionado em uma outra placa fora da aeronave. O circuito para medição dos ângulos de ataque longitudinal e lateral também será posicionado fora da placa.
- Para medição de velocidade é utilizado um tubo de Pitot (acoplado ao sensor de pressão diferencial), posicionado na direção longitudinal da aeronave;
- Os sinais de sensores não serão amplificados ou atenuados. Embora talvez seja interessante fazê-lo, isso não será considerado no atual projeto.

5.3.1. Circuitos para os Sensores

- Sensores de Pressão:

Os circuitos mostrados nas figuras 5.10 e 5.11 são realizados desta forma por recomendação do fabricante (Freescle). A idéia dos capacitores é desacoplar os CIs em termos de ruído elétrico, e amenizar ruídos inseridos na linha devido ao sensor.

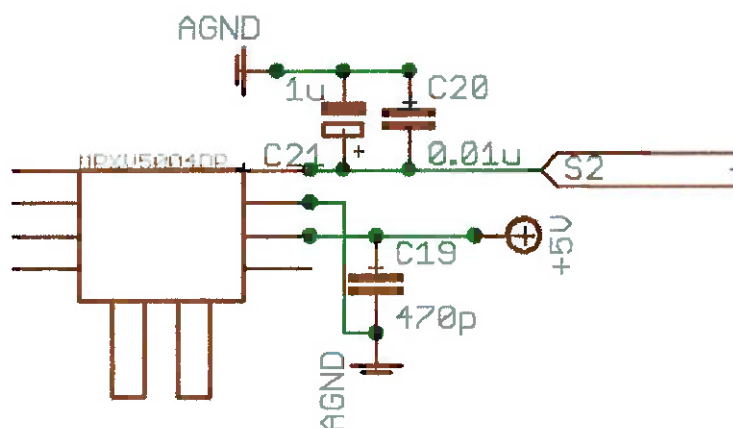


Figura 5.10 – Circuito para sensor de pressão diferencial (analógico)

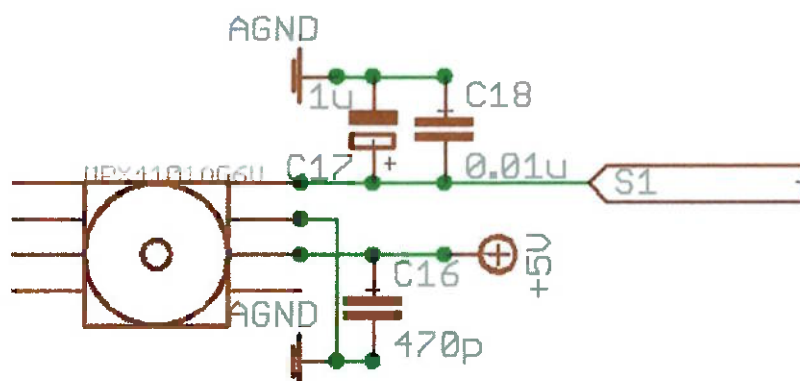


Figura 5.11 – Circuito para sensor de pressão barométrica absoluta (analógico)

- Sensor de Umidade:

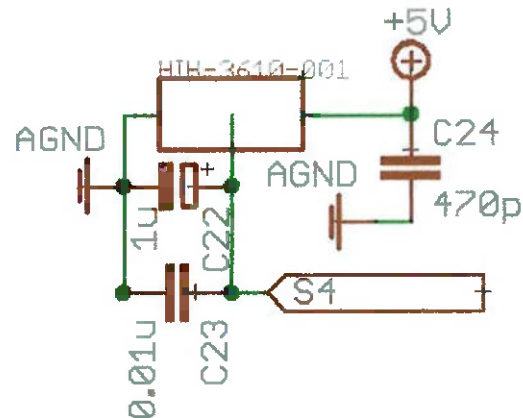


Figura 5.12 – Circuito para sensor de umidade (analógico)

- Sensor de Temperatura:

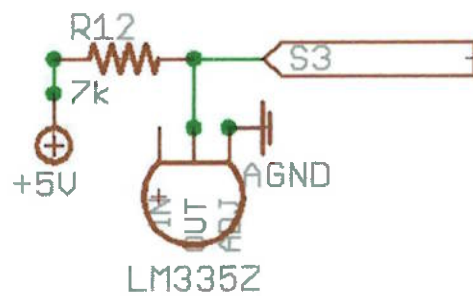


Figura 5.13 – Circuito para sensor de temperatura (analógico)

- Sensor de Ângulo de Ataque Lateral e Longitudinal:

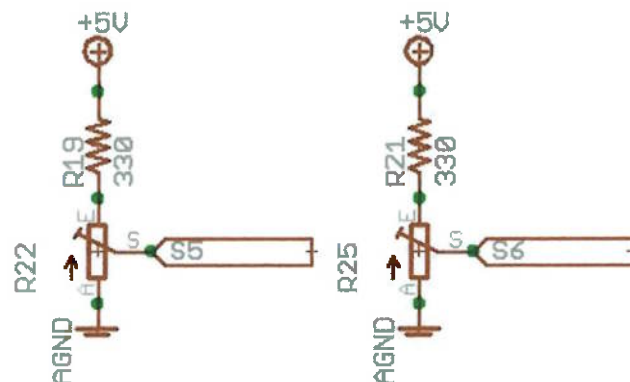


Figura 5.14 – Circuito para sensor de ângulo de ataque longitudinal e lateral

- Medidor de Rotação:

Como explicado no item 5.2.1, o medidor de rotação é um circuito que emite um pulso para cada rotação da hélice.

Como pode ser visualizado na figura 5.15, o LED emite os pulsos em uma frequência conhecida, dada pelo circuito oscilador (canto direito inferior da figura). Estes pulsos são percebidos no foto-diodo (infravermelho) devido à reflexão da onda eletromagnética no adesivo refletor colado na hélice. O circuito do canto esquerdo superior é um filtro, diminuindo assim interferência externa, mesmo no espectro infravermelho. O circuito do canto direito superior emite os pulsos digitais, que serão contados no contador do MCU. O PIC167F887A tem dois temporizadores / contadores, um de 8 bits outro de 16 bits. Será utilizado para o medidor de rotação o contador de 16 bits, pois o contador de 8 bits será utilizado na interrupção, como descrito no capítulo 6.

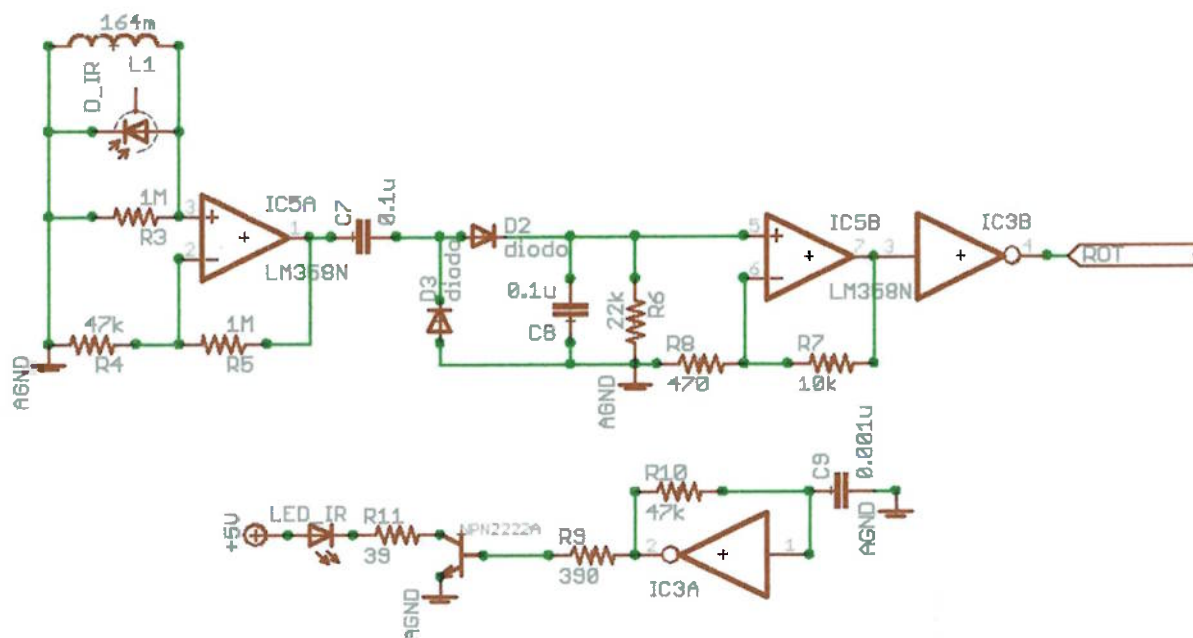


Figura 5.15 – Medidor de rotação

5.3.2. Circuito para o MMC

Como dito no item 5.2.3, o MMC trabalha com lógica CMOS, desta forma necessita-se de um circuito divisor resistivo para compatibilizar os níveis HIGH TTL e HIGH CMOS. Um outro aspecto importante não mencionado anteriormente, é que o MMC da SanDisk só pode ser gravado em pacotes múltiplos de 512 bytes. Porém o PIC 16F877A só possui 368 bytes de RAM. Desta forma, os dados podem ser gravados deixando metade do pacote de dados em vazio, portanto perdendo 50% da capacidade de armazenamento do MMC, ou pode-se utilizar uma memória serial como registrador, armazenando o pacote de dados, a ser lido e armazenado seqüencialmente no MMC. Essa segunda opção é adotada, utilizando-se a memória serial I2C 24LC04B da Microchip (anexo C). Como o PIC não possui duas portas seriais SPI/I2C, então o protocolo de comunicação com esta memória será implementado por software.

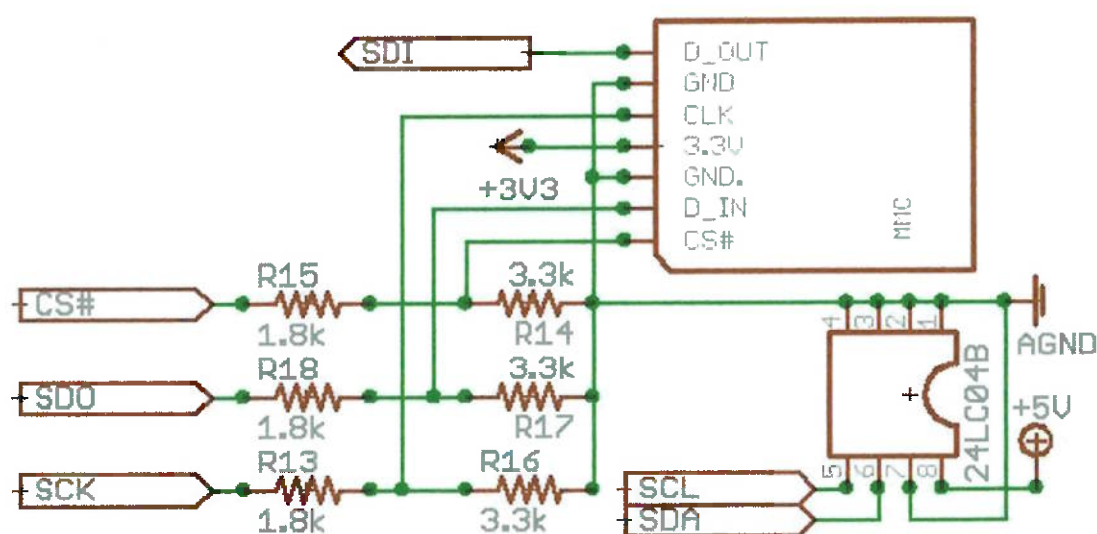


Figura 5.16 – Circuito para o MMC

5.3.3. Circuito de Alimentação

Necessita-se, nesta placa, de uma fonte de 5 V e outra de 3,3 V (para o MMC). Como a corrente drenada é estimada em 1,5 A, uma única fonte é supostamente suficiente, já que o regulador LM7805 fornece até 1,5 A, e o LM1117 fornece até 0,8 A. O *pack* de baterias utilizado, como mencionado no item 5.1, apresenta no mínimo 7 V após uma hora de utilização (um dia de voo, ou seja, seis vôos de dez minutos cada), fornecendo durante todo tempo 1,5 A.

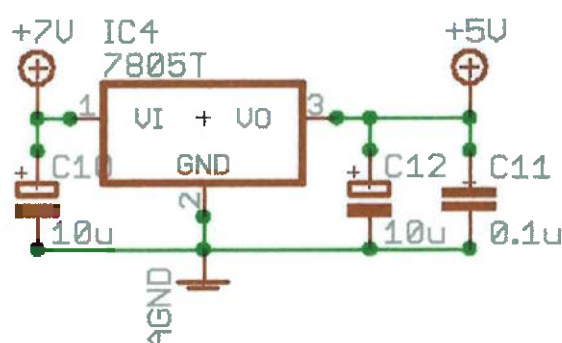


Figura 5.17 – Circuito para fonte de 5V

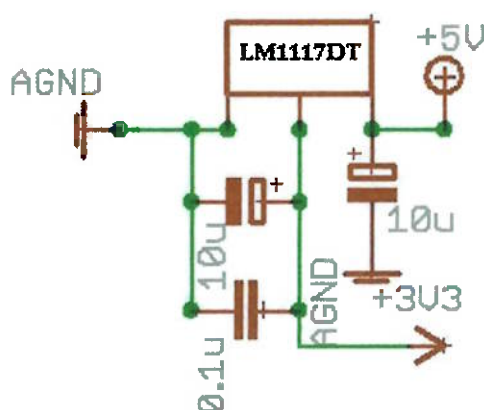


Figura 5.18 – Circuito para fonte de 3,3V

5.3.4. Circuito para Comunicação Externa

Como mencionado no item 5.1, a placa sendo projetada deve ser capaz de comunicar-se serialmente com um PC, e ainda deve possuir um barramento para comunicação externa com outra placa, visando uma futura expansão do sistema. Para a comunicação serial, usa-se um conversor de nível (HIN232, similar ao MAX232) de TTL para RS232. Para controle do fluxo pelo barramento paralelo, usa-se o CI 74LS245 (*Bus Transceiver* com controle *tri-state*).

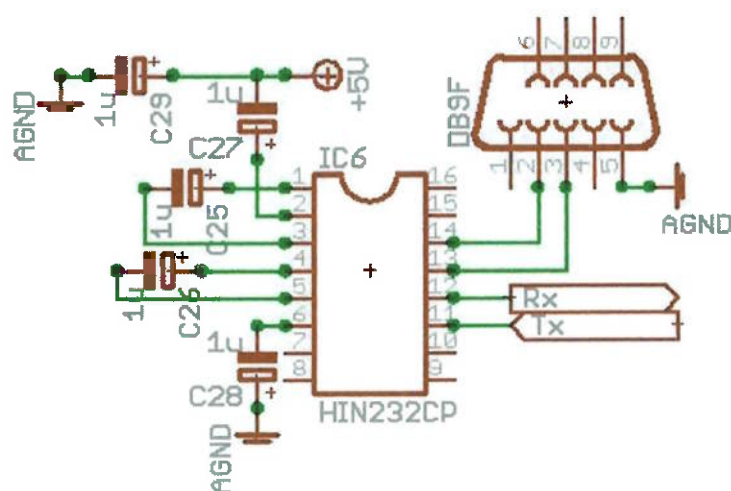


Figura 5.19 – Circuito para comunicação serial com PC

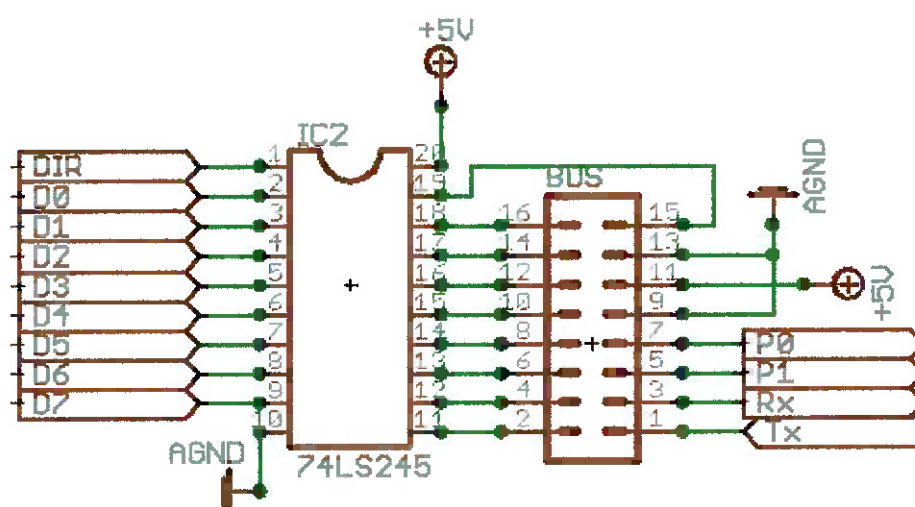


Figura 5.20 – Circuito para comunicação paralela, visando futura expansão do sistema

5.3.5. Circuito para o Microcontrolador

Embora se necessite de um clock de 1 MHz, como definido no item 5.1, adotaremos um cristal de 20 MHz (máxima frequência que o PIC16F877A pode trabalhar). Para esta frequência, o circuito do oscilador deve seguir a configuração HS (*High Speed*), de acordo com o fabricante – Microchip, sendo os valores de capacitância desse circuito também dados pelo fabricante.

O pino RB3/PGM é utilizado para entrar no modo de programação, sendo parte do recurso de LV ICSP (*Low Voltage In Circuit Serial Programming*) do PIC16F877A. Para entrar no modo programação deve-se levar esse pino de LOW para HIGH, depois levar o pino MCLR# de LOW para HIGH (e manter).

O pino MCLR# é o *hard reset* do PIC, sendo sensível à alta corrente e ruído. Desta forma, o circuito de alimentação desse pino ilustrado, é recomendado pelo fabricante, para que não ocorra nenhuma falha no sistema.

Os pinos RA2 e RA3, podem ser utilizados como referência positiva e negativa das entradas analógicas, porém serão utilizados como entradas analógicas normais.

Serão utilizados três LEDs (pinos 15, 19, e 20) para visualizar o *status* do sistema. Serão utilizados os pinos 21 e 22, com chaves para interface com o usuário, possibilitando iniciar a aquisição e iniciar comunicação serial com PC.

O botão soldado ao MCLR# é o *hard reset*, sendo a única forma de re-iniciar o sistema.

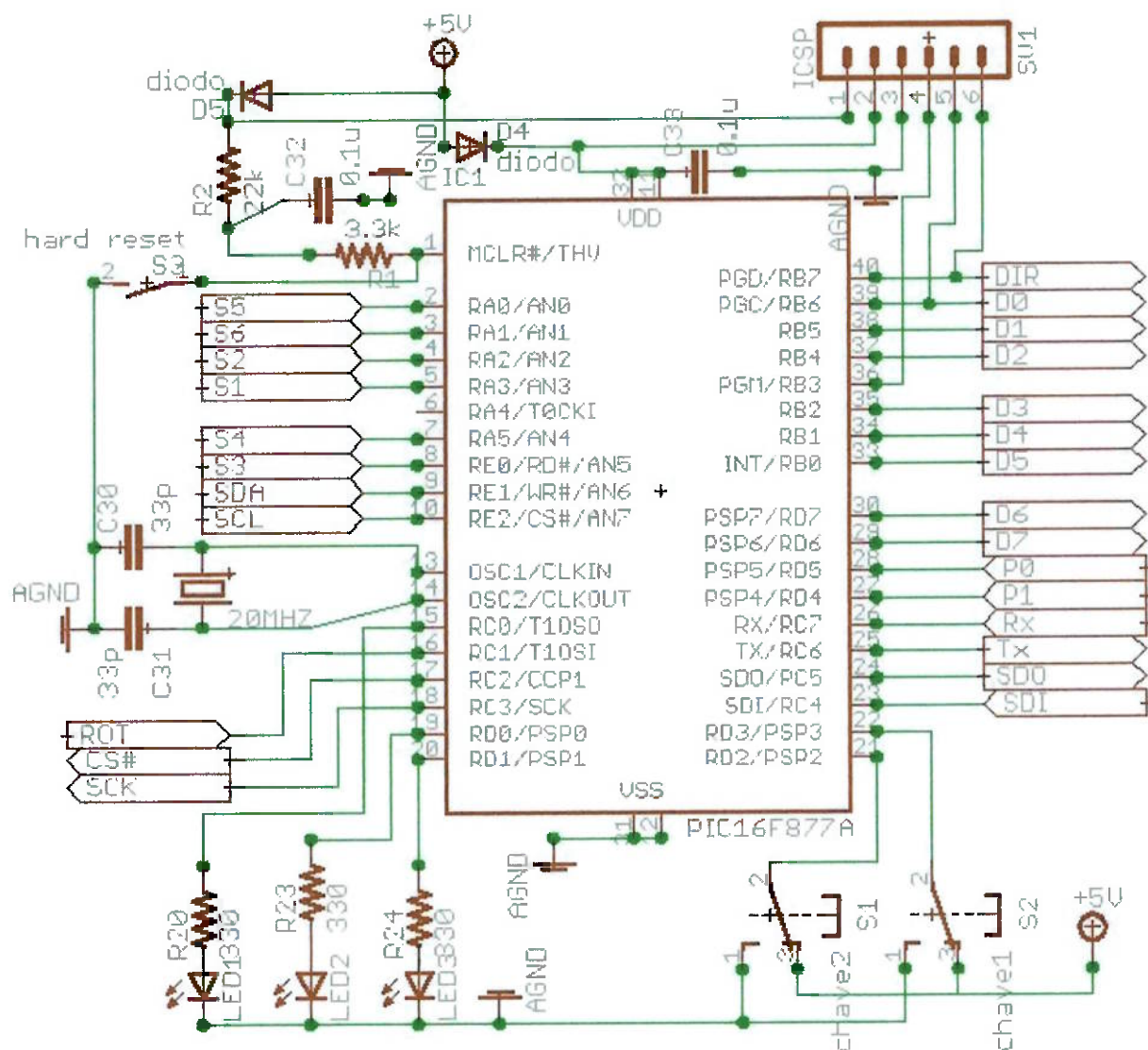


Figura 5.21 – Circuito do microcontrolador

5.4. Análise

Neste item serão mostradas, com base nas informações fornecidas pelos fabricantes, a corrente drenada e a potência dissipada máximas, assim como a corrente de operação medida.

A tabela 5.4 é mostrada considerando sempre o maior caso de corrente drenada e potência dissipada dos componentes mencionados. É considerada a corrente drenada não somente pela alimentação do circuito, mas também pelos pinos de entrada.

Tabela 5.4 – Corrente drenada e potência dissipada

Componente	Fabricante	Corrente Drenada (mA)	Potência Dissipada (mW)
PIC16F877A	Microchip	1050	10000
24LC04B	Microchip	15	60
MPXH6101A6U	Freescall	10	50
MPXV5004DP	Freescall	10	50
HIH-3610-001	Honeywell	0,3	1,5
LM335Z	National Semiconductor	15	50
BPV10NF	VISHAY	50	220
TSUS4300	TEMIC	200	170
MMC	SanDisk	60	240
74LS245B1	ST Microelectronics	290	1000
HIN232CP	Intersil	150	850
LM358N	ST Microelectronics	55	500
CD40106BCN	National Semiconductor	16	700
LM7805*	ST Microelectronics	2200	15000
LM1117DT*	National Semiconductor	1500	10000
NPN2222A	General Semiconductor	60	625
LEDS**	Kingbright	100	480

* Carga drenando maior corrente admissível

** Soma de todos. Considerando resistor de 330 Ohms no catodo.

No total temos, na condição de maior dissipação a 25°C, corrente drenada máxima é de 2,1 A. O regulador LM7805 não consegue fornecer essa corrente, sem risco de ruptura. Os teste

indicaram que a corrente drenada operacional é menor que 0,1 A, muito abaixo da corrente máxima drenada, e da corrente estimada.

Com esta nova estimativa de corrente máxima drenada, o circuito de alimentação deverá, em aplicações futuras, ser modificado como mostrado na figura 5.22. Devido ao fato do MCU drenar quase metade da corrente consumida, então será utilizado um regulador exclusivo para o PIC. Ainda neste projeto o circuito mostrado na figura 5.22 não será utilizado, pois os testes indicaram que a princípio essa modificação não é necessária.

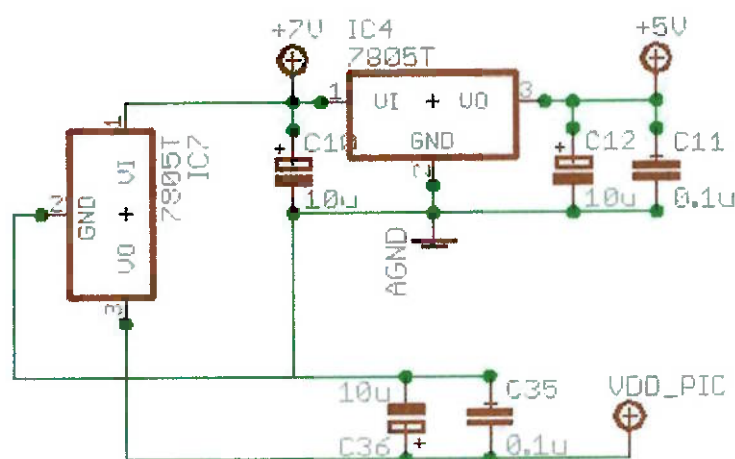


Figura 5.22 – Circuito modificado para fonte de 5V

5.5. Lista de Materiais e Orçamento Final

Em um primeiro momento a idéia foi terceirizar a manufatura da placa, porém, devido ao alto custo desse serviço, o mesmo foi realizado pelo autor. Uma placa completa, e uma outra incompleta e simples (para testes de software – capítulo 7) foram manufaturadas. Além disso, foi comprado material extra para se fazer outra placa. A tabela 5.5 ilustra o orçamento final.

Tabela 5.5 – Materiais e Orçamento Final

Componente	Qtde.	Custo (R\$)	Material	Qtde.	Custo (R\$)
PIC16F877A	2	30,00	Solda (22g)	5	3,50
24LC04B	4	8,00	Resistor	150	0,25
29F040	3	15,00	Capacitor	100	0,35
NPN2222A	5	2,50	Chave	8	2,50
HIH-3610-001	2	135,00	Potenciômetro	4	5,00
LM335Z	3	4,00	Conector	20	0,75
BPV10NF	5	2,00	Fiação	8m	1,50/m
TSUS4300	5	2,00	Bobina	6	10,00
MMC	2	75,00	Botão	6	2,00
74LS245B1	3	2,50	Cristal	2	1,50
HIN232CP	3	3,50	Soquete	11	3,50
LM358N	5	2,50	Diodo	20	0,20
CD40106BCN	3	2,50	Zener	10	0,20
LM7805	5	1,50	LED	20	0,20
LM1117DT	3	3,50	Placa fenolite	4	8,50
Pilha	8	3,50	Suporte pilha	4	3,50

Foram gastos no total aproximadamente R\$1015,00. A equipe Poli Active TAM, como parceira, contribuiu com cerca de 35% do valor total.

Capítulo 6

Projeto do *Firmware*

Neste capítulo será definido todo o *software* embarcado. O projeto do *software* não segue nenhuma metodologia específica, por se tratar de um *firmware* relativamente simples.

Toda a diagramação do *software* foi realizada em UML e fluxogramas. O diagramador UML adotado foi o Pacestar UML Diagrammer versão 4.19 da Pacestar Software. A linguagem de programação adotada foi a linguagem C. Vários compiladores foram analisados (cc5x, EWPIC, MikroC, etc.), e o compilador utilizado foi o MikroC versão 2.1.0.0 da MikroElektronika. As vantagens deste compilador são suas bibliotecas e seus cabeçalhos (para a maioria dos microcontroladores PIC) já prontos para aplicação, além de suas ferramentas de análise e sua interface amigável. A versão *shareware* deste *software* limita o uso de memória *flash* do MCU em 2k (assim como todos os compiladores amigáveis para PIC). Esta limitação tornou-se um problema, pois o sistema na sua versão final superava tal limite. Desta forma o *firmware* foi modificado e simplificado, perdendo, portanto, um pouco de sua robustez. A descrição das bibliotecas do MikroC, e o código fonte encontram-se no Anexo E.

6.1. Diagrama de Estados

O sistema em questão funcionará da seguinte forma:

- Será efetuada a configuração dos registradores do PIC, de forma que este opere como se deseja (pinos de entrada, saída, etc);
- Os LEDs piscarão ao mesmo tempo quatro vezes. Somente após esta etapa o sistema irá atuar conforme a configuração das chaves de interface com usuário;
- Se ambas as chaves estiverem abertas, o circuito permanecerá em *stand by* (sem amostrar os dados, nem armazenar na memória). Nesse caso, somente o LED um estará aceso;
- Se somente a chave um estiver fechada, o circuito deverá amostrar, e jogar os dados na memória, e somente o LED dois permanecerá aceso;
- Se somente a chave dois estiver fechada, o circuito não amostrará os dados, apenas irá ler a memória e jogar os dados da memória para a serial RS232 (comunicação externa). Nesse caso somente o LED três estará aceso;
- Se ambas as chaves estiverem fechadas, o circuito amostrará os dados, armazenará na memória, e enviará para a serial RS232, e ambos LEDs dois e três permanecerão acesos;
- Se os três LEDs estiverem acesos ou apagados significa que houve algum erro;
- O sistema poderá ser re-iniciado somente via *hard reset* (botão de *reset* no pino MCLR#), ou obviamente no instante de troca de baterias;
- O usuário pode mudar a configuração das chaves a qualquer instante;
- O circuito poderá ser programado e re-programado via LVP (ICSP), bastando para isso conectar o circuito em um chip de programação nas saídas LVP0 até LVP5. Quando for programado desta forma, a alimentação do circuito deve ser cessada, pois o chip de programação fornecerá a alimentação necessária.
- Não será utilizado nenhum *timer* externo.

Em cada estado um outro estado de verificação das chaves será acessado. O diagrama de estados (padrão UML - figura 6.1) define o *software* a ser implementado. Em futuras aplicações este *software* será modificado, possibilitando a expansão do sistema.

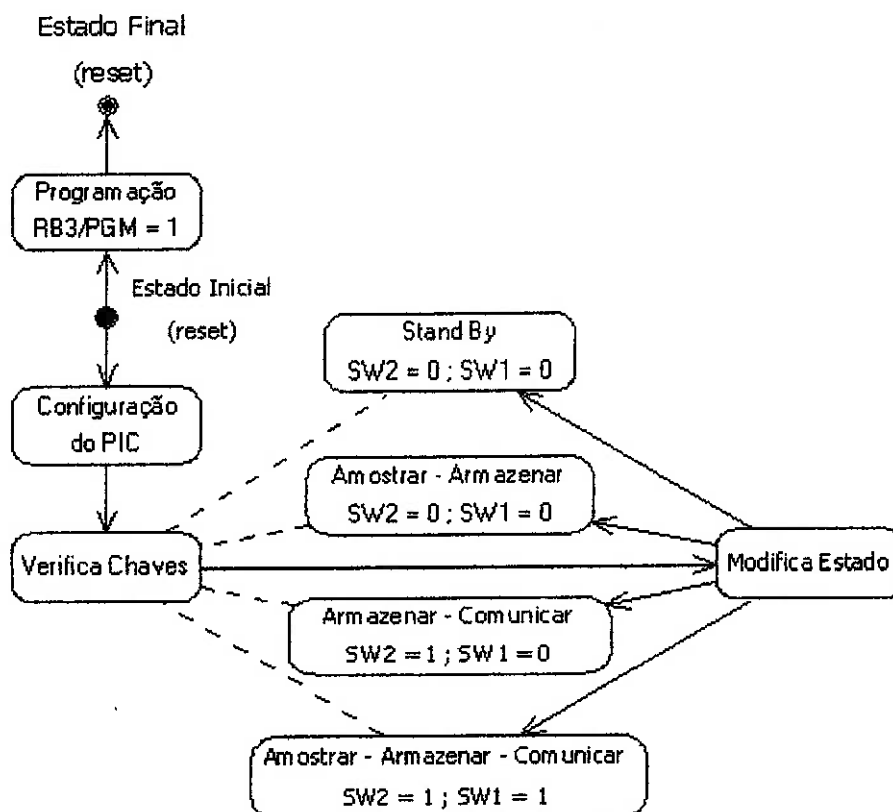


Figura 6.1 – Diagrama de estados

6.2. Configuração do PIC

Para funcionamento conforme esperado, o PIC (assim como qualquer outro MCU) deve ter seus registradores configurados. Essa configuração define se um pino será utilizado como entrada ou saída, se as interrupções estão ativadas ou não, se a porta analógica está ativada ou não, se o contador está ativado ou não, etc.

No presente projeto precisa-se configurar os registradores da porta A (entradas analógicas), da porta B (I/Os para o barramento, e pino de programação PGM), porta C (contador, SPI, e USART), e porta D (saída para LEDs, entrada para chaves, e I/Os para o barramento). Além disso, deve-se configurar os registradores de interrupção, *timer*, e configuração geral do PIC (*reset*, *sleep*, *watch dog*, etc).

A tabela 6.1 ilustra os registradores que serão configurados e suas funções. No anexo D encontra-se todo o mapa de registradores do MCU.

Tabela 6.1 – Registradores configurados

Registrador	Função
TRISX	Controle direcional dos pinos da porta X
PORTX	Registrador de entrada/saída da porta X
ADCON0	Registrador de controle do conversor AD
ADCON1	Registrador de controle do conversor AD
SSPCON	Registrador de controle da porta serial SSP
INTCON	Controle/ <i>status</i> das interrupções
PIE	Controle/ <i>status</i> das interrupções
PIR	Controle/ <i>status</i> das interrupções
OPTION_REG	Controle/ <i>status</i> das interrupções e dos <i>timers</i>
TICON	Controle do <i>timer0</i>
TSR	Controle/ <i>status</i> da porta USART
SSPSR	Controle/ <i>status</i> da porta SPI
SPBRG	Controle da taxa de transferência da USART
SSPSTAT	<i>Status</i> da porta SPI
TXSTA	<i>Status</i> da transmissão da USART
RCSTA	<i>Status</i> da recepção da USART
ADRESH	Saída do conversor AD
ADRESL	Saída do conversor AD
SSPBUF	Saída da SPI
TXREG	<i>Buffer</i> de transmissão da USART
RCREG	<i>Buffer</i> de recepção da USART
TMR1H	Saída do <i>timer1</i>
TMR1L	Saída do <i>timer1</i>

Neste instante verifica-se claramente a vantagem do compilador utilizado: suas bibliotecas têm funções que configuram a maioria desses registradores.

6.3. Descrição do *firmware*

Antes de iniciar a leitura desse item, recomenda-se a leitura do código fonte, anexo E.

O sistema, ao iniciar, declara as variáveis e funções, configura o PIC, e pisca os LEDs quatro vezes. Em seguida o sistema opera baseado nas duas chaves de interface com usuário, assim como descrito em 6.1. No modo *stand by*, ambas as chaves estão abertas, e o LED 1 está aceso. Nesse caso, o programa principal chama a função F_StandBy. Enquanto a chave não for modificada, essa função permanecerá sendo executada, apenas chamando a função F_LeChaves, que atualiza o valor da variável global “chaves”.

Caso o usuário modifica a configuração das chaves, de modo que somente a chave um esteja fechada, então o programa pára de executar a função F_StandBy e entra na função F_AmostraArmazenaComunica, de onde sairá, se e somente se, a configuração das chaves for modificada. Dentro dessa função o sistema executa (função F_CarregaBuffer) a leitura dos sensores e carrega o buffer (variável global “buffer_aux”). O *buffer* é uma matriz quadrada de variáveis do tipo “*unsigned short*” (8bits) com oito linhas, ou seja, 512bits.

Para não ocorrer *overflow* de RAM, o buffer é escrito na memória 24LC04B via I2C (protocolo realizado por software, funções Soft_I2C). Para não ocorrer “*end of file*” no MMC, é utilizado uma variável global “endereço” (*unsigned long*). Após amostrar e armazenar na memória I2C oito vezes o “buffer_aux” (ou seja, 512bytes, tamanho necessário para escrever no MMC), esta função lê a memória 24LC04, e envia os dados para o MMC, e caso a chave dois também esteja fechada, então os dados serão enviados também para a USART.

Caso o usuário modifique as chaves, de modo somente a chave dois esteja fechada, então o sistema irá entrar para executar a função `F_LeMmcComunica`. Esta função lê o MMC, deste seu endereço zero, e envia os dados para a USART.

O maior tempo de desenvolvimento do firmware foi despendido para implementação de funções do MMC (já que o MikroC não possui bibliotecas para comunicação de MMC com PIC16F877A), e configuração do PIC.

Um ponto importante na implementação do software é verificar e respeitar o protocolo de comunicação da memória e do MMC. A memória utiliza um protocolo I2C padrão (sinais de *clock*, e *data*). As figuras 6.2 e 6.3 ilustram o protocolo em leitura e escrita de um único bloco. A memória também possui capacidade de escrita e leitura de blocos múltiplos.

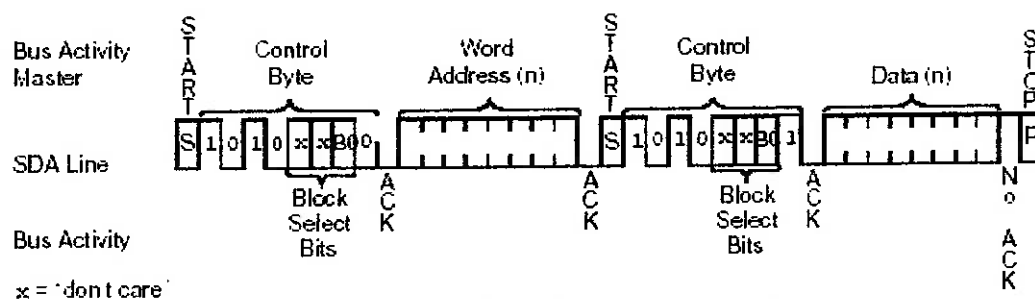


Figura 6.2 – Leitura de um byte na memória I2C

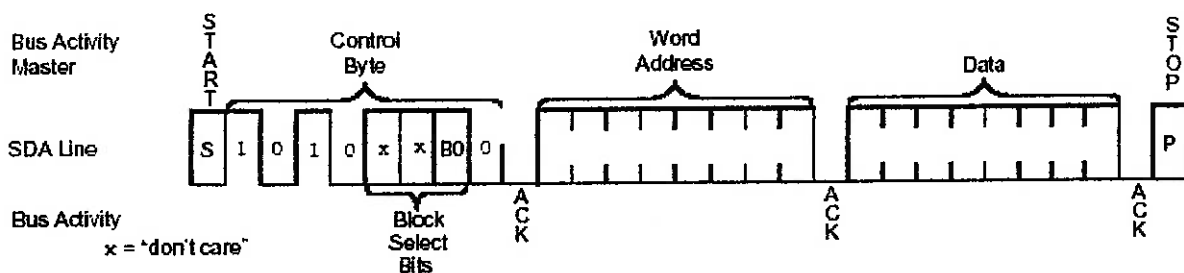


Figura 6.3 – Escrita de um byte na memória I2C

O cartão de memória possui dois modos de operação, modo MMC e modo SPI. Além disso, o MMC pode ser acessado, usando-se diferentes estruturas de arquivo (FAT com *boot*, sem *boot*, partição desconhecida, formato universal, etc). Para este projeto adota-se o modo SPI, com partição em formato universal (*raw*). O protocolo SPI identifica somente o meio física (*data in*, *data out*, *clock*, e *chip select*), porém não identifica o protocolo de comunicação em termos de software propriamente dito. O órgão que gerencia as especificações de MMC é a Associação de Cartões de Multimedia (www.mmca.org). O manual completo de especificações pode ser comprado via *web*, no valor de U\$500.00! O fabricante SanDisk disponibiliza um *datasheet* razoável no *site* para OEM (*Original Equipment Manufacturer*), www.sandisk.com/OEM. As figuras 6.4, e 6.5 ilustram o protocolo para leitura e escrita de um bloco único no MMC. Outras características importantes no MMC são os seus registradores de configuração, seus comandos (o MMC possui 64 comandos), seu ciclo de *reset* e *power on*. O anexo C contém informações adicionais sobre o MMC e a memória.

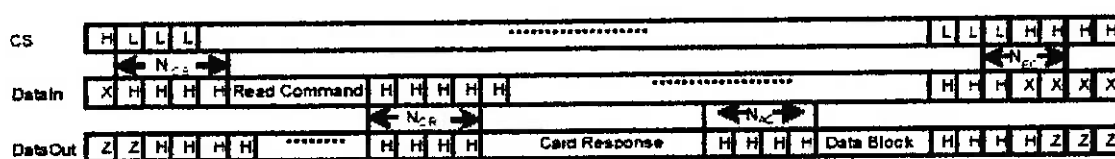


Figura 6.4 – Leitura de um byte no MMC

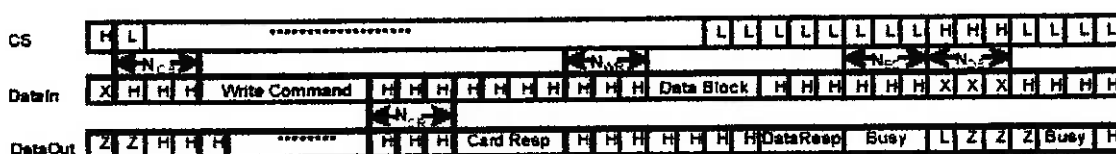


Figura 6.5 – Escrita de um byte no MMC

6.4. Programação do PIC

Microcontroladores PIC podem ser acessados no modo programação de duas formas:

- Serial, aplicando 12V no pino MCLR#, enquanto os pinos RB6 e RB7 permanecem aterrados;
- Serial, aplicando 5V ao pino RB3, e em seguida aplicando 5V ao pino MCLR#. Este modo chama-se *Low Voltage Programming* e somente pode ser utilizado caso o bit LVP da palavra de configuração esteja configurado como "1".

Após acessar o modo de programação, os dados são enviados serialmente, pinos RB7 (*data*) e RB6 (*clock*). A tabela 6.2 resume os pinos utilizados e suas funções.

Tabela 6.2 – Pinos utilizados na programação

Pin Name	During Programming		
	Function	Pin Type	Pin Description
RB3	PGM	I	Low voltage ICSP programming input if LVP configuration bit equals '1'
RB6	CLOCK	I	Clock input
RB7	DATA	I/O	Data input/output
MCLR	VTEST MODE	P*	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

Legend: I = Input. O = Output. P = Power

- * To activate the Programming mode, high voltage needs to be applied to the MCLR input. Since MCLR is used for a level source, this means that MCLR does not draw any significant current.

Após entrar no modo de programação, a mesma deve ser efetuada de acordo com a sequência abaixo.

1. Carregar dados no endereço atual, usando o comando “*Load Data*”;
2. Enviar comando “*Increment Address*”;
3. Carregar dados no endereço atual, usando o comando “*Load Data*”;
4. Repetir passos “2” e “3” por seis vezes;
5. Enviar comando “*Begin Programming*”;
6. Esperar tempo de programação (1 ms);
7. Enviar comando “*End Programming*”;
8. Incrementar endereço enviando o comando “*Increment Address*”;
9. Repetir esta sequência de acordo com a necessidade da programação;

Os possíveis comandos no modo de programação são resumidos na tabela 6.3.

Tabela 6.3 – Comandos no modo de programação

Command	Mapping (MSB ... LSB)					Data	Voltage Range
Load Configuration	0	0	0	0	0	0, data (14), 0	2.2V - 5.5V
Load Data for Program Memory	0	0	0	1	0	0, data (14), 0	2.2V - 5.5V
Read Data from Program Memory	0	0	1	0	0	0, data (14), 0	2.2V - 5.5V
Increment Address	0	0	1	1	0		2.2V - 5.5V
Begin Erase/Programming Cycle	0	1	0	0	0	4 ms typical, internally timed	2.2V - 5.5V
Begin Programming Only Cycle	1	1	0	0	0	1 ms typical, externally timed	4.5V - 5.5V
Bulk Erase Program Memory	0	1	0	0	1	4 ms typical, internally timed	4.5V - 5.5V
Bulk Erase Data Memory	0	1	0	1	1	4 ms typical, internally timed	4.5V - 5.5V
Chip Erase	1	1	1	1	1	4 ms typical, internally timed	4.5V - 5.5V
Load Data for Data Memory	0	0	0	1	1	0, data (14), 0	2.2V - 5.5V
Read Data from Data Memory	0	0	1	0	1	0, data (14), 0	2.2V - 5.5V
End Programming	1	0	1	1	1		

Capítulo 7

Ensaio e Testes de Protótipos

Com o intuito de certificar o funcionamento do *software*, e do *hardware*, foram realizados testes de comunicação com PC, escrita e leitura na memória 24LC04B, leitura das portas analógicas, leitura e escrita em portas digitais.

Para realizar os primeiros testes, foi desenvolvido um circuito simples em uma placa de fenolite preparada. Esse circuito possibilitava a comunicação via USART com o PC, a escrita e a leitura da memória 24LC04B, a leitura e a escrita em portas digitais, com LEDS de sinalização e botão de interface com usuário. A figura 7.1 ilustra uma foto do circuito teste em questão. A idéia desse circuito era certificar o software a ser utilizado na placa final, e também a configuração do *hardware*, já que os circuitos da memória, da USART, do PIC, dos LEDS, e dos botões são idênticos aos utilizados nessa placa teste.

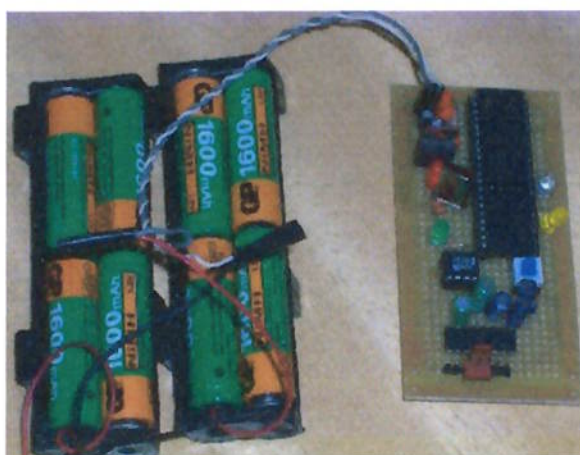


Figura 7.1 – Circuito teste em funcionamento

No primeiro teste o MCU foi programado com um *firmware* que acendia ou apagava os LEDS, de acordo com o status do botão. Esse teste foi realizado com sucesso.

No segundo teste, o PIC recebia um byte do tipo “char” via USART (57600 kbps, sem paridade, sem CR/LF, e com um *stop* bit) do PC, enviava de volta o mesmo byte, e piscava os LEDS duas vezes caso a operação finalizasse com sucesso. Para comunicação com o PIC foi utilizado o *software* RComSerial (www.rogercom.com). Este teste foi realizado com sucesso.

No próximo teste, caso o botão não estivesse pressionado, o sistema permanecia piscando todos os LEDS ao mesmo tempo. Caso o botão fosse pressionado, os LEDS piscariam um seguido do outro, o sistema executaria a leitura das portas analógicas com uma taxa de 50Hz, e enviaria os sinais para a USART. Esse teste foi realizado com sucesso.

O quarto teste visava testar a leitura e escrita na memória 24LC04B. Foi verificado que os resistores de *pullup* utilizados para a memória estavam com errados. Após a troca desses resistores o quarto teste foi finalizado com sucesso.

Posteriormente foram realizados testes com o protótipo da placa final, aquela manufaturada pelo autor.

No primeiro teste dessa placa foram verificados seis curtos. O pequeno espaçamento entre as trilhas, e a péssima qualidade da solda e da placa de fenolite adquiridas foram as principais causas para esses curtos. Após corrigir este problema a placa foi testada novamente, porém sem sucesso. A hipótese inicial era regulador queimado, pois a tensão estava regulada em 4,6V e não 5,1V como de usual. O regulador foi trocado, e foi inserido um LED de sinalização de alimentação. Ainda assim a placa não respondeu a nenhum estímulo. O microcontrolador foi testado, para averiguar se este tinha sido danificado, mas ele estava em perfeitas condições. As trilhas de alimentação, terra, e MCLR# foram testadas com multímetro, e estavam corretas. Foi

verificada a configuração dos pinos no MCU, de modo a averiguar se algum pino de entrada estava sendo utilizado como saída, ou vice-versa. Nenhum problema foi encontrado.

Testes realizados, juntamente com o Prof. Rui Baptista na BLUE TEC, indicaram que o problema encontrava-se no circuito do oscilador e no pino RB3. O capacitor utilizado no circuito oscilador estava, por erro de montagem, com valor errado e não recomendado pelo fabricante. O pino RB3, como informado no capítulo 5, é o pino de programação LVP. A variação de nível neste pino leva o PIC a entrar ou sair do modo de programação, desta forma, esse pino deve ser aterrado com um *pulldown*. Ainda na BLUE TEC foram realizadas medições de corrente e frequência de oscilação. A corrente de operação drenada é inferior a 100 mA. Foi também verificado que o regulador está aquecendo-se rapidamente, e em futuras aplicações deverá ser implementado um sistema de refrigeração.

Após corrigir os problemas citados, foram realizados os últimos testes de comunicação e armazenamento de dados. Estes testes foram realizados com sucesso.

O protótipo final pode ser visualizado na figura 7.2.

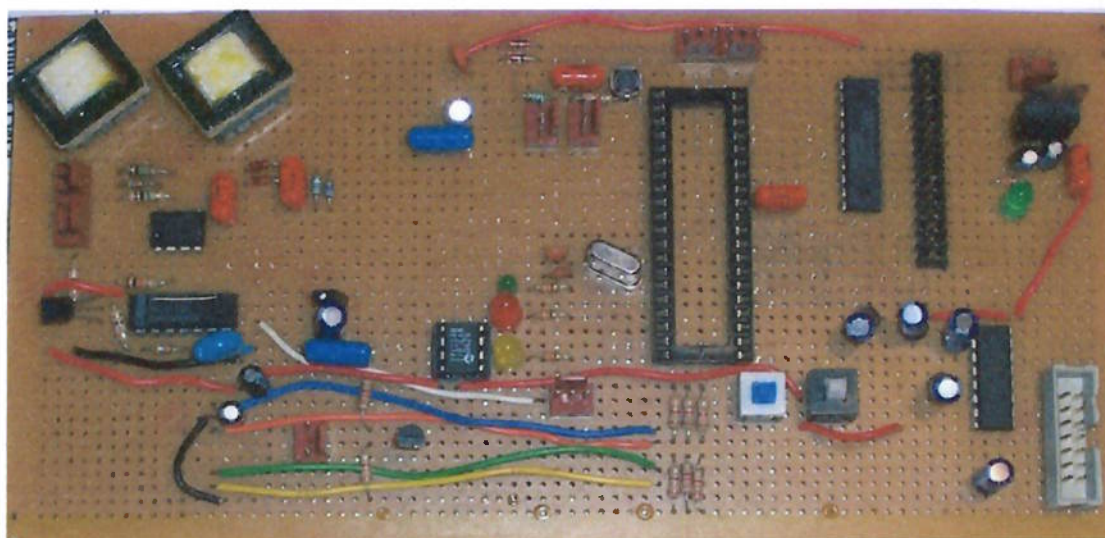


Figura 7.2 – Protótipo final

Capítulo 8

Conclusões

São apresentadas neste capítulo conclusões relativas:

- Aos sensores adotados;
- Ao MCU escolhido;
- Ao uso de MMC;
- Ao layout e arquitetura do *hardware*;
- Ao *firmware* desenvolvido e compilador utilizado;
- Aos testes realizados.

Ao final é apresentada uma conclusão geral, que considera também o ponto de vista de aprendizagem.

- Sensores

Os sensores de umidade HIH-3610-001, pressão barométrica MPXH6101A6U, temperatura LM335Z, circuitos para medição de rotação da hélice, ângulo de ataque longitudinal e lateral, são muito simples e eficientes para esta aplicação. O sensor mais oneroso foi o HIH. Analisando esse conjunto de sensores conclui-se que, para essa aplicação, o mesmo pode ser classificado como boa escolha, a menos do sensor de pressão diferencial MPXV5004DP, que pode, em futuras aplicações, ser substituído pelo DC001NGC4 (da Honeywell).

Os sinais dos sensores não foram tratados por nenhum filtro, porém, em aplicações futuras deve-se considerar a equalização, amplificação ou tratamento dos dados mensurados.

- Microcontrolador

O PIC16F877A foi escolhido, por três razões principais: ele satisfazia os requisitos, sem nenhuma folga, ou seja, ele poderia ser usado de maneira a empregar quase 100% dos seus recursos (a menos da memória de dados EEPROM e *flash*, que dependeriam da complexidade do *firmware*); ele é usualmente utilizado em projetos de eletrônica, portanto de fácil reposição e fácil obtenção de material didático e de referência; ele possui um baixo custo. Embora, a comunicação com o MMC tenha ficado prejudicada, devido à restrição de RAM deste MCU, a solução utilizando uma memória serial externa, pode ser implementada, trazendo apenas dificuldade para o projeto do *software*. Um PIC18, por exemplo, poderia ter sido escolhido, desta forma o *firmware* seria muito mais simples, não haveria necessidade de utilizar memória serial (como a utilizada 24LC04B), porém o custo seria quase 50% maior, e muitos recursos do PIC18 permaneceriam ociosos.

- Memória

A escolha pelo uso do MMC foi muito satisfatória do ponto de vista do *hardware*, pois o circuito ficou extremamente simples, e com uma grande capacidade de armazenamento, a uma taxa de transferência muito maior que a necessária. Do ponto de vista do *software*, a complexidade do protocolo de comunicação do MMC atrasou em uma semana o andamento do projeto, pois tal complexidade não havia sido prevista no cronograma. A complexidade

mencionada diz respeito ao nível de detalhes e configurações, e não ao tamanho do código fonte, já que o código final não ultrapassou 300 linhas úteis, ocupando aproximadamente 2k de *flash*. O maior problema dessa solução diz respeito à vida útil da memória e do MMC, pois memórias EEPROM e *flash* têm uma vida útil (em ciclos de escrita e leitura) muito curta, desta forma, deve-se estudar em futuras aplicações, a possibilidade de utilização de memória RAM.

- *Layout e Arquitetura do Hardware*

A placa protótipo possui as dimensões de 10 cm x 15 cm x 3,5 cm. Essas medidas estão próximas das medidas definidas em requisitos. O *layout* projetado, embora um tanto complexo, (com dupla face e muitas trilhas longas e próximas), é factível. No caso de terceirizar a manufatura da próxima placa, esse layout pode ser reduzido para 10 cm x 10 cm x 3,5 cm. Em análise global, o *layout* atual pode ser melhorado, mas está relativamente bom.

A arquitetura projetada, visando a expansão do sistema, com comunicação serial e paralela, é muito utilizada, por exemplo, em PC/104, onde os módulos deste se comunicam via barramentos paralelos e seriais. Essa arquitetura, embora satisfaça os requisitos e tenha um comportamento confiável, deve ser sempre revisada, pois o número de possíveis arquiteturas é muito elevado, e várias soluções são factíveis (conforme capítulo 4).

- *Firmware* Desenvolvido e Compilador Utilizado

O *software* desenvolvido é relativamente simples, principalmente devido ao fato de o compilador utilizado (MikroC) conter diversas bibliotecas prontas. O *firmware* em sua primeira versão era robusto, de modo a verificar possíveis falhas de comunicação com memória / MMC, e

reiniciar a transmissão / recepção dos dados. Porém, devido à restrição de limitação do código do programa na versão *shareware* do compilador, essa robustez teve que ser sacrificada, sendo o *software* reduzido a 70% do seu tamanho original.

Outros compiladores foram testados, como o EWPIC, e o cc5x. O cc5x embora muito divulgado e largamente utilizado, possui poucas bibliotecas, uma interface ultrapassada, e poucas ferramentas de análise. O EWPIC possui uma interface melhor que a do cc5x, mas inferior a do MikroC. Além disso, possui recursos de análises, porém a complexidade do compilador era um fator limitante, devido a restrições de tempo de desenvolvimento do *firmware*, e o MikroC é uma ferramenta muito fácil de se utilizar, portanto, a melhor escolha dentre as opções.

- Testes

Durante os testes realizados foram encontradas falhas como curto-circuito, resistores e capacitores errados, e configuração errada do pino de programação (RB3). Todos estes problemas foram resolvidos. O sistema em sua versão final é capaz de ler os sensores, e armazenar na memória ou enviar para o PC com sucesso. Devido às restrições de tempo não foram realizados testes de calibração com os sensores.

Assim, conclui-se que o desenvolvimento desse trabalho proporcionou uma grande agilidade para o entendimento de sistemas embarcados, uma rapidez notável na análise de especificações e *datasheet*, e um conhecimento amplo do quadro atual de projetos de veículos autônomos. Também proporcionou a ampliação de conhecimentos em eletrônica, no uso de microcontroladores, cartões de memória, baterias, sensores, etc. Além disso, as restrições de tempo levaram o autor a buscar por soluções rápidas e eficientes.

REFERÊNCIAS

- [1] Nelson, Robert C.. Flight Stability and Automatic Control. McGraw-Hill Book Company, New York, 1989.
- [2] Raymer, D. P.. Aircraft Design: A conceptual Approach. 2ª. ed.. American Institute of Aeronautics and Astronautics, Washington, DC, 1992.
- [3] Ogata, K.. Engenharia de Controle Moderno. 4ª. ed.. Pearson Prentice Hall, São Paulo, 2003.
- [4] USAF Stability and Control DATCOM. Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, 1959.
- [5] Dorf, Richard C.. Sistemas de Controle Moderno. LTC, São Paulo, 2001.
- [6] Catálogo Eletrônico disponível em <<http://black.rc.unesp.br/balthazar/voo/dan2.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [7] Catálogo Eletrônico disponível em <<http://sunvalley.stanford.edu/~jsjang/movies/AIAA035592.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [8] Catálogo Eletrônico disponível em <http://www.cis.upenn.edu/departamental/reports/PennUAV_techreport.pdf>. Último acesso em 10 de fevereiro de 2006.
- [9] Messina, Paolo. Building a Platform for UAV Research. Departament of Electrical Systems and Automation, University of Pisa, Italy, 2003.
- [10] Catálogo Eletrônico disponível em <<http://www.et.byu.edu/~dbk4/papers/BeardEtAl04.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [11] Catálogo Eletrônico disponível em <http://www.mil.ufl.edu/publications/fcrar03/Grzywna%202003_gps.pdf>. Último acesso em 10 de fevereiro de 2006.
- [12] Catálogo Eletrônico disponível em <http://www.ae.gatech.edu/people/ejohnson/uav2004_fcs20.pdf>. Último acesso em 10 de fevereiro de 2006.
- [13] Catálogo Eletrônico disponível em <<http://www.seas.upenn.edu/~pappasg/papers/CDC04-UAV.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [14] Catálogo Eletrônico disponível em <<http://www.seas.ucla.edu/coopcontrol/kickoff/chichka.pdf>>. Último acesso em 10 de fevereiro de 2006.

- [15] Catálogo Eletrônico disponível em <http://clubs.engr.arizona.edu/arc/0405site/TEAM_PAPERS/UniversityofArizona2004.pdf>. Último acesso em 10 de fevereiro de 2006.
- [16] Catálogo Eletrônico disponível em <<http://sunvalley.stanford.edu/~jsjang/movies/AIAA035592.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [17] Catálogo Eletrônico disponível em <http://www.eecs.berkeley.edu/~tah/Publications/from_control_models_to_real-time_code_using_giotto.pdf>. Último acesso em 10 de fevereiro de 2006.
- [18] Catálogo Eletrônico disponível em <<http://www.eecs.berkeley.edu/~ergen/docs/p22-koo.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [19] Catálogo Eletrônico disponível em <http://www.ece.uwaterloo.ca/~warg/media/papers/WARG_2001.pdf>. Último acesso em 10 de fevereiro de 2006.
- [20] Catálogo Eletrônico disponível em <http://www.ene.unb.br/~gaborges/pesquisa/uav/heli1/relatorio_projeto-final_heli1.pdf>. Último acesso em 10 de fevereiro de 2006.
- [21] Catálogo Eletrônico disponível em <http://cres.usc.edu/pubdb_html/files_upload/384.pdf>. Último acesso em 10 de fevereiro de 2006.
- [22] Catálogo Eletrônico disponível em <<http://rhw.staff.ul.ie/omerdic/My%20Papers/ICRODIC2004.PDF>>. Último acesso em 10 de fevereiro de 2006.
- [23] Catálogo Eletrônico disponível em <http://murx.medien.ifi.lmu.de/~albrecht/pdf/icmipui_2003.pdf>. Último acesso em 10 de fevereiro de 2006.
- [24] Catálogo Eletrônico disponível em <http://mechatronics.eng.buffalo.edu/research/smartcar/Publications/DETC2003_CIE48258.pdf>. Último acesso em 10 de fevereiro de 2006.
- [25] Catálogo Eletrônico disponível em <[tp://www.igvc.org/deploy/design/reports/dr37.pdf](http://www.igvc.org/deploy/design/reports/dr37.pdf)>. Último acesso em 10 de fevereiro de 2006.
- [26] Catálogo Eletrônico disponível em <<http://dsplabs.cs.utt.ro/~micha/publications/pdfs/MiniRobot.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [27] Catálogo Eletrônico disponível em <http://lis.epfl.ch/publications/roggend_spiking.pdf>. Último acesso em 10 de fevereiro de 2006.
- [28] Catálogo Eletrônico disponível em <http://www.radiig.informatik.tu-muenchen.de/people/schroetd/Research/Publications/diploma_thesis.pdf>. Último acesso em 10 de fevereiro de 2006.
- [29] Catálogo Eletrônico disponível em <<http://www.gel.usherbrooke.ca/laborius/papers/IROS00eme.pdf>>. Último acesso em 10 de fevereiro de 2006.

- [30] Catálogo Eletrônico disponível em <<http://i61www.ira.uka.de/users/asfour/publications/IROS-2004-1.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [31] Catálogo Eletrônico disponível em <<http://152.99.129.29/its/cdrom/3008.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [32] Catálogo Eletrônico disponível em <<http://www.cs.cmu.edu/~arvinds/pubs/conro.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [33] Catálogo Eletrônico disponível em <<http://polaris.dit.upm.es/~jpunte/strl/transparencias/Espacio.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [34] Catálogo Eletrônico disponível em <<http://waas.stanford.edu/~www/papers/gps/PDF/Thesis/JDGautierThesis03.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [35] Catálogo Eletrônico disponível em <<http://www.mil.ufl.edu/publications/fcrar03/Plew.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [36] Catálogo Eletrônico disponível em <http://www.c-lab.de/data/Publication_DB/DB/Koch_Markus_00/paperkorukl_aim2003_last_final.pdf>. Último acesso em 10 de fevereiro de 2006.
- [37] Catálogo Eletrônico disponível em <http://www.galaxyscientific.com/agingaircraft2002/SESSIONS/4/4C3_JOHNSON_PPT.PDF>. Último acesso em 10 de fevereiro de 2006.
- [38] Catálogo Eletrônico disponível em <http://www.omg.org/news/meetings/workshops/RT_2003_Manual/Presentations/6-2_Aslam-Mir_etal2.pdf>. Último acesso em 10 de fevereiro de 2006.
- [39] Catálogo Eletrônico disponível em <<http://www.mini.pw.edu.pl/~matysiak/af/AF05.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [40] Catálogo Eletrônico disponível em <<http://www.acq.osd.mil/dsb/reports/uav.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [41] Catálogo Eletrônico disponível em <http://www.dglr.de/veranstaltungen/archiv/2004_uav-ucaf-mav/12-Stuetz-Udovic/dglr_2004uav_Hochfliegendes-UAV.pdf>. Último acesso em 10 de fevereiro de 2006.
- [42] Catálogo Eletrônico disponível em <<http://www.csse.monash.edu.au/~carlo/archive/PAPERS/ADFA/ucav-2001.pdf>>. Último acesso em 10 de fevereiro de 2006.
- [43] Catálogo Eletrônico disponível em <http://www.lsa.isep.ipp.pt/publications/presentations/aula_uav.pdf>. Último acesso em 10 de fevereiro de 2006.
- [44] Portal Teses e Dissertações da USP <<http://www.teses.usp.br>>. Documento Eletrônico ARARA_KleberMt.pdf. Projeto Arara. Último acesso em 10 de fevereiro de 2006.

[45] Catálogo Eletrônico disponível em <<http://black.rc.unesp.br/balthazar/voo/danilo1.pdf>>. Último acesso em 10 de fevereiro de 2006.

[46] Catálogo Eletrônico disponível em <<http://black.rc.unesp.br/balthazar/voo/danilo2.pdf>>. Último acesso em 10 de fevereiro de 2006.

[47] Catálogo Eletrônico disponível em <<https://dspace.mit.edu/bitstream/1721.1/16662/1/56549421.pdf>>. Último acesso em 10 de fevereiro de 2006.

[48] Portal Teses e Dissertações da USP <<http://www.teses.usp.br>>. Documento Eletrônico ARARA_Diss.pdf. Projeto Arara. Último acesso em 10 de fevereiro de 2006.

[49] Portal Teses e Dissertações da USP <<http://www.teses.usp.br>>. Documento Eletrônico ARARA_AUVIS.pdf. Projeto Arara. Último acesso em 10 de fevereiro de 2006.

[50] Okamoto Jr., Jun. Metodologia de Projeto, Escola Politécnica da USP, São Paulo, 2004.

[51] Cipelli, Antonio Marcos V.; Markus, Otávio; Sandrini, Waldir. Teoria e Desenvolvimento de Projetos de Circuitos Eletrônicos, 19ª. ed.. Ed. Érica, São Paulo, 2004.

ANEXO A – Sensores

A1. Sensor de Pressão MPXV5004DP (Freescale)

Characteristic	Symbol	Min	Typ	Max	Units
Pressure Range	P_{OP}	0	—	3.92 400	kPa mm H ₂ O
Supply Voltage ⁽¹⁾	V_S	4.75	5.0	5.25	V _{DC}
Supply Current	I_S	—	—	10	mA _{DC}
Span at 305 mm H ₂ O (3 kPa) ⁽²⁾	V_{FSS}	—	3.0	—	V
Offset ⁽³⁾ (4)	V_{OFF}	0.75	1.0	1.25	mV
Sensitivity	V/P	—	1.0 9.8	—	V/kPa mV/mm H ₂ O
Accuracy ⁽⁵⁾	—	—	—	±1.5 ±2.5	% V_{FSS} % V_{FSS}

1. Device is ratiometric within this specified excitation range.
2. Span is defined as the algebraic difference between the output voltage at specified pressure and the output voltage at the minimum rated pressure.
3. Offset (V_{OFF}) is defined as the output voltage at the minimum rated pressure.
4. Accuracy (error budget) consists of the following:
 - Linearity: Output deviation from a straight line relationship with pressure over the specified pressure range.
 - Temperature Hysteresis: Output deviation at any temperature within the operating temperature range, after the temperature is cycled to and from the minimum or maximum operating temperature points, with zero differential pressure applied.
 - Pressure Hysteresis: Output deviation at any pressure within the specified range, when this pressure is cycled to and from the minimum or maximum rated pressure at 25 °C.
 - Offset Stability: Output deviation, after 1000 temperature cycles, -30 to 100 °C, and 1.5 million pressure cycles, with minimum rated pressure applied.
 - TcSpan: Output deviation over the temperature range of 10 to 60 °C, relative to 25 °C.
 - TcOffset: Output deviation with minimum rated pressure applied, over the temperature range of 10 to 60 °C relative to 25 °C.
 - Variation from Nominal: The variation from nominal values, for Offset or Full Scale Span, as a percent of V_{FSS} , at 25 °C.
5. Auto Zero at Factory Installation: Due to the sensitivity of the MPXV5004G, external mechanical stresses and mounting position can affect the zero pressure output reading. Autozeroing is defined as storing the zero pressure output reading and subtracting this from the device's output during normal operations. Reference AN1636 for specific information. The specified accuracy assumes a maximum temperature change of ±5 °C between autozero and measurement.

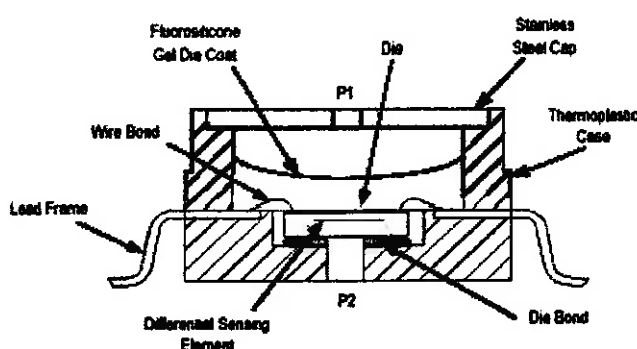


Figure 2. Cross-Sectional Diagram (Not to Scale)

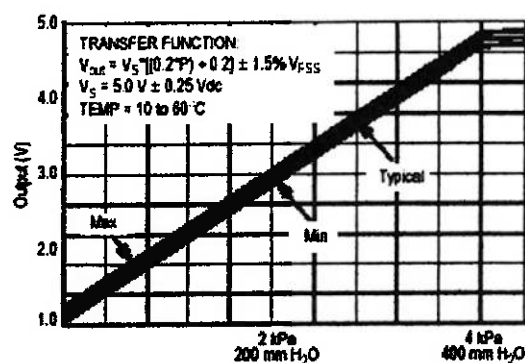
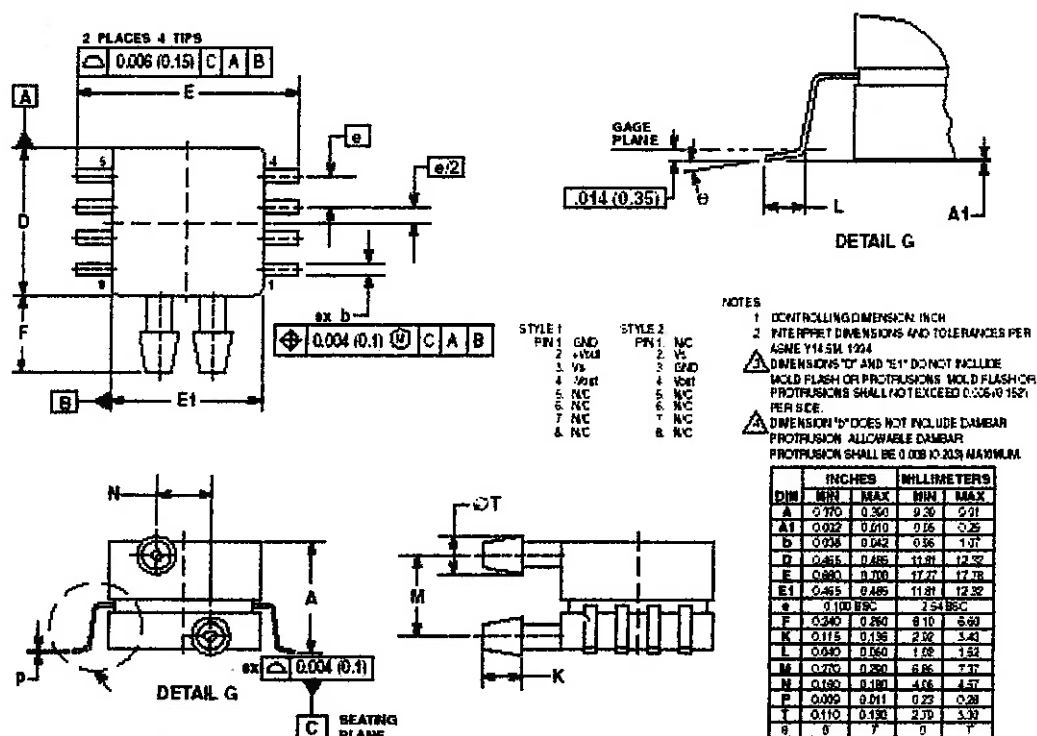
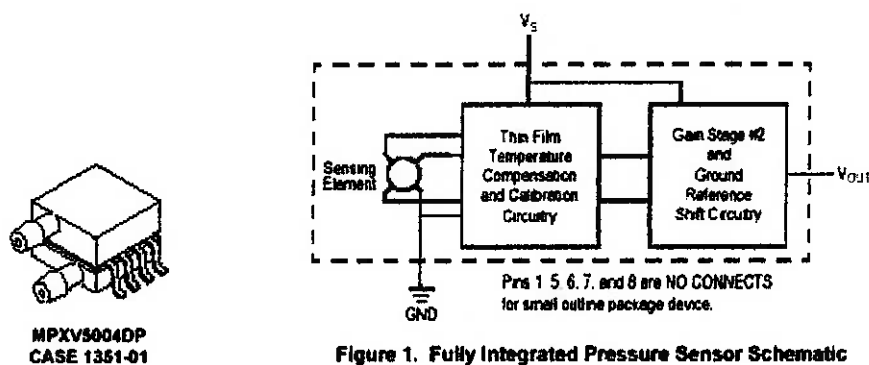


Figure 4. Output versus Pressure Differential
(See Note 5 in Operating Characteristics)

Table 1. Maximum Ratings⁽¹⁾

Rating	Symbol	Value	Unit
Maximum Pressure (P1 > P2)	P _{MAX}	16	kPa
Storage Temperature	T _{STG}	-30 to +100	C
Operating Temperature	T _A	0 to +85	C

1. Exposure beyond the specified limits may cause permanent damage or degradation to the device.



CASE 1351-01
ISSUE 0
SMALL OUTLINE PACKAGE
SURFACE MOUNT

A2. Sensor de Pressão MPX4101AC6U (Freescale)

Table 2. Operating Characteristics ($V_S = 5.1$ Vdc, $T_A = 25^\circ\text{C}$ unless otherwise noted, $P_1 > P_2$.
Decoupling circuit shown in Figure 3 required to meet electrical specifications.)

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range ⁽¹⁾	P_{OP}	15	—	102	kPa
Supply Voltage ⁽²⁾	V_S	4.85	5.1	5.35	Vdc
Supply Current	I_o	—	7.0	10	mAdc
Minimum Pressure Offset @ $V_S = 5.1$ Volts ⁽³⁾ (0 to 85 °C)	V_{off}	0.171	0.252	0.333	Vdc
Full Scale Output @ $V_S = 5.1$ Volts ⁽⁴⁾ (0 to 85 °C)	V_{FSO}	4.870	4.951	5.032	Vdc
Full Scale Span @ $V_S = 5.1$ Volts ⁽⁵⁾ (0 to 85 °C)	V_{FSS}	—	4.7	—	Vdc
Accuracy ⁽⁶⁾ (0 to 85 °C)	—	—	—	± 1.72	% V_{FSS}
Sensitivity	V/P	—	54	—	mV/kPa
Response Time ⁽⁷⁾	t_R	—	15	—	ms
Output Source Current at Full Scale Output	I_{OS}	—	0.1	—	mAdc
Warm-Up Time ⁽⁸⁾	—	—	20	—	ms
Offset Stability ⁽⁹⁾	—	—	± 0.5	—	% V_{FSS}

1. 10 kPa (kilopascal) equals 0.145 psi

2. Device is ratiometric within this specified excitation range.

3. Offset (V_{off}) is defined as the output voltage at the minimum rated pressure.

4. Full Scale Output (V_{FSO}) is defined as the output voltage at the maximum or full rated pressure.

5. Full Scale Span (V_{FSS}) is defined as the algebraic difference between the output voltage at full rated pressure and the output voltage at the minimum rated pressure.

6. Accuracy (error budget) consists of the following:

- Linearity: Output deviation from a straight line relationship with pressure over the specified pressure range.
- Temperature Hysteresis: Output deviation at any temperature within the operating temperature range, after the temperature is cycled to and from the minimum or maximum operating temperature points, with zero differential pressure applied.
- Pressure Hysteresis: Output deviation at any pressure within the specified range, when this pressure is cycled to and from the minimum or maximum rated pressure, at 25 °C.
- T_C Span: Output deviation over the temperature range of 0 to 85 °C, relative to 25 °C.
- T_C Offset: Output deviation with minimum rated pressure applied, over the temperature range of 0 to 85 °C, relative to 25 °C.
- Variation from Nominal: The variation from nominal values, for Offset or Full Scale Span, as a percent of V_{FSS} , at 25 °C.

7. Response Time is defined as the time for the incremental change in the output to go from 10% to 90% of its final value when subjected to a specified step change in pressure.

8. Warm-up Time is defined as the time required for the product to meet the specified output voltage after the Pressure has been stabilized.

9. Offset Stability is the product's output deviation when subjected to 1000 hours of Pulsed Pressure, Temperature Cycling with Bias Test.

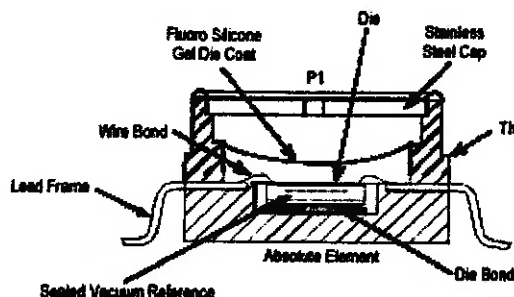


Figure 2. Cross Sectional Diagram SSOP (not to scale)

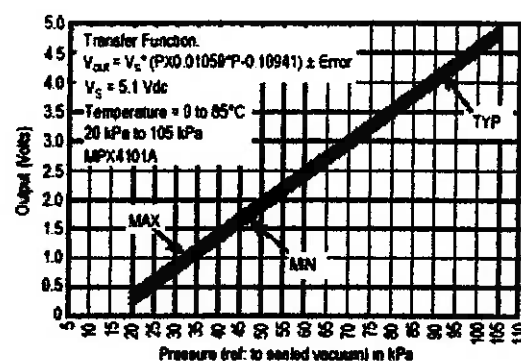
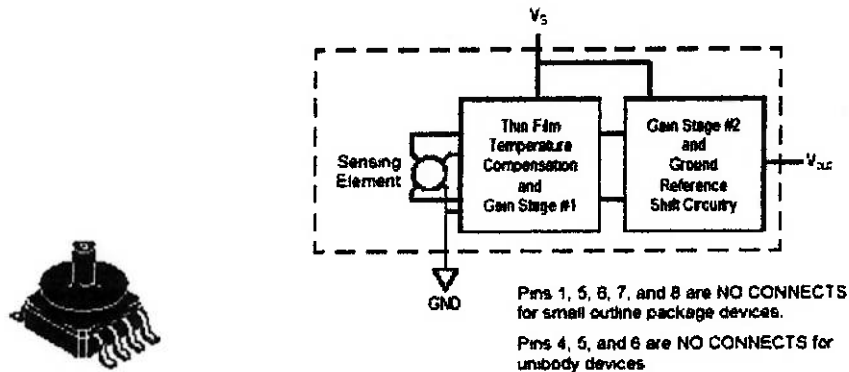


Figure 4. Output versus Absolute Pressure

Table 1. Maximum Ratings⁽¹⁾

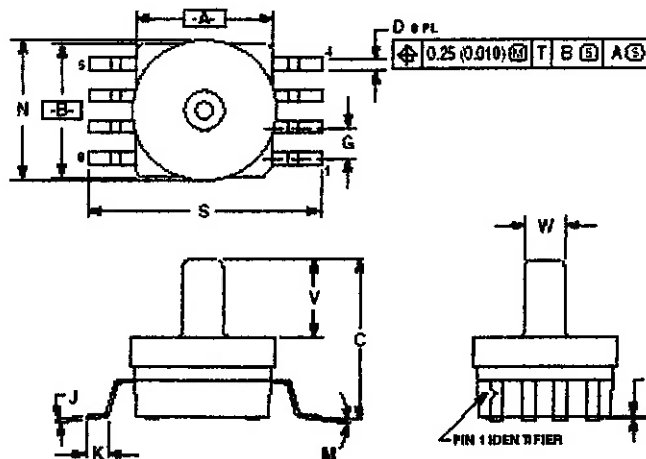
Rating	Symbol	Value	Unit
Maximum Pressure (P1 > P2)	P _{MAX}	400	kPa
Storage Temperature	T _{STG}	-40 to +125	°C
Operating Temperature	T _A	-40 to +125	°C

1. Exposure beyond the specified limits may cause permanent damage or degradation to the device.



MPXA4101AC6U
CASE 482A-01

Figure 1. Temperature Compensated and Calibrated Pressure Sensor Schematic



NOTES
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M-1982
2. CONTROLLING DIMENSION INCH
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006)
5. ALL VERTICAL SURFACES 5° TYPICAL DRAFT

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.415	0.425	10.54	10.70
B	0.415	0.425	10.54	10.70
C	0.900	0.920	22.75	23.21
D	0.268	0.272	6.80	6.97
E	0.100 BSC		2.54 BSC	
F	0.002	0.010	0.05	0.25
G	0.000	0.011	0.23	0.28
H	0.051	0.071	1.29	1.80
I	0	0	0	0
J	0.444	0.448	11.26	11.39
K	0.750	0.755	18.91	19.14
L	0.245	0.255	6.22	6.44
M	0.115	0.125	2.92	3.17

CASE 482A-01
ISSUE A
SMALL OUTLINE PACKAGE

A3. Sensor de Umidade HIH-3610-001 (Honeywell)

TABLE 1: PERFORMANCE SPECIFICATIONS

Parameter	Condition
RH Accuracy ⁽¹⁾	±2% RH, 0-100% RH non-condensing, 25 °C, V _{supply} = 5 Vdc
RH Interchangeability	±5% RH, 0-60% RH; ±8% @ 90% RH typical
RH Linearity	±0.5% RH typical
RH Hysteresis	±1.2% RH span maximum
RH Repeatability	±0.5% RH
RH Response Time, 1/e	15 sec in slowly moving air at 25 °C
RH Stability	±1% RH typical at 50% RH in 5 years
Power Requirements	
Voltage Supply	4 Vdc to 5.8 Vdc, sensor calibrated at 5 Vdc
Current Supply	200 µA at 5 Vdc
Voltage Output	V _{out} = V _{supply} (0.0062(Sensor RH) + 0.16), typical @ 25 °C (Data printout option provides a similar, but sensor specific, equation at 25 °C.)
V _{supply} = 5 Vdc	0.8 Vdc to 3.9 Vdc output @ 25 °C typical
Drive Limits	Push/pull symmetric; 50 µA typical, 20 µA minimum, 100 µA maximum Turn-on ≤ 0.1 sec
Temperature Compensation	True RH = (Sensor RH)/(1.093-0.0012T), T in °F True RH = (Sensor RH)/(1.0546-0.00216T), T in °C
Effect @ 0% RH	±0.007 %RH/°C (negligible)
Effect @ 100% RH	-0.22% RH/°C (<1% RH effect typical in occupied space systems above 15 °C (59 °F))
Humidity Range	
Operating	0 to 100% RH, non-condensing ⁽¹⁾
Storage	0 to 90% RH, non-condensing
Temperature Range	
Operating	-40 °C to 85 °C (-40 °F to 185 °F)
Storage	-51 °C to 125 °C (-60 °F to 257 °F)
Package ⁽²⁾	Three pin, solderable SIP in molded thermoset plastic housing with thermoplastic cover
Handling	Static sensitive diode protected to 15 kV maximum

Notes:

1. Extended exposure to ≥90% RH causes a reversible shift of 3% RH.
2. This sensor is light sensitive. For best results, shield the sensor from bright light.

FIGURE 1: RH SENSOR CONSTRUCTION

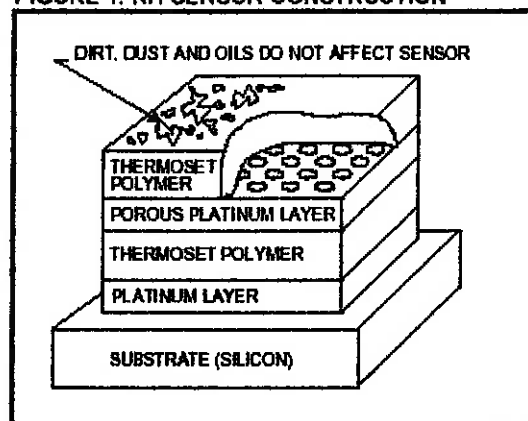


FIGURE 3: OUTPUT VOLTAGE VS RELATIVE HUMIDITY AT 0 °C, 25 °C, 85 °C

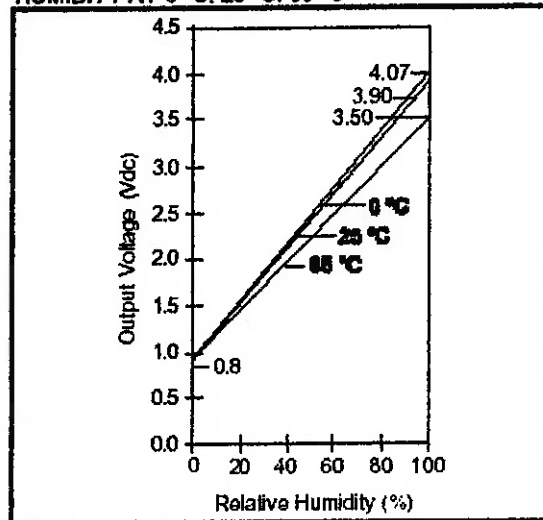
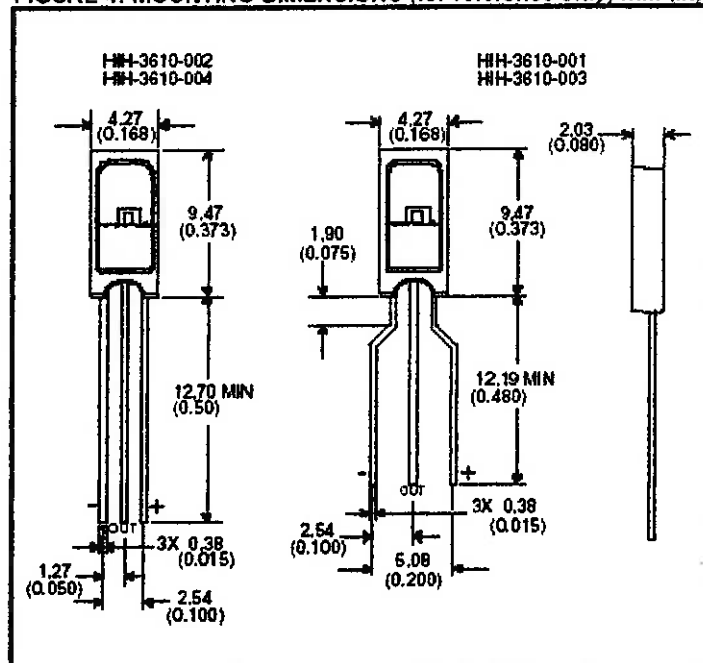


FIGURE 4: MOUNTING DIMENSIONS (for reference only) mm (in)



A4. Sensor de Temperatura LM335Z (National Semiconductor)

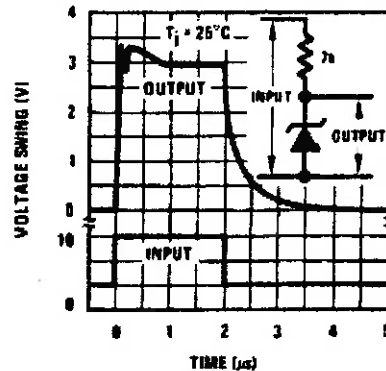
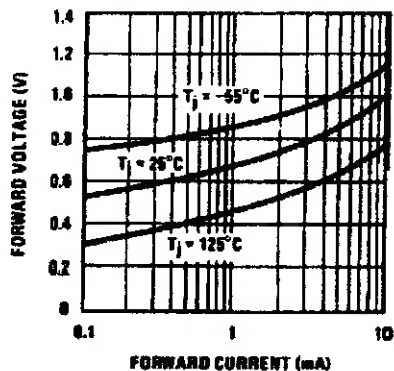
Temperature Accuracy (Note 1)

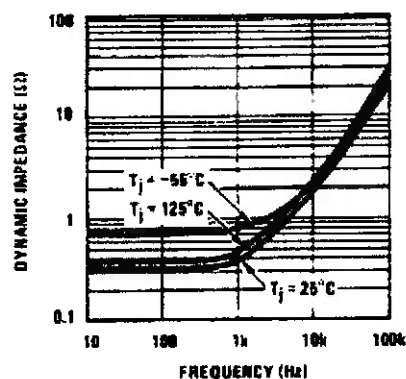
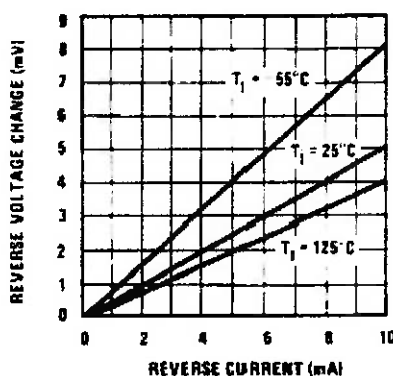
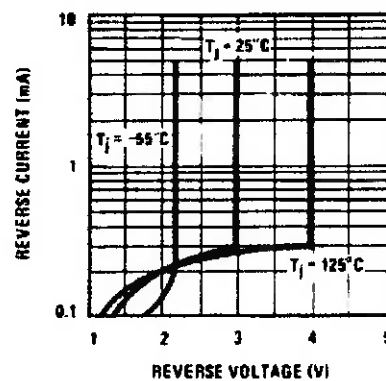
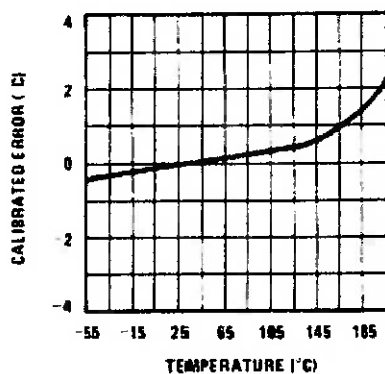
LM335, LM335A

Parameter	Conditions	LM335A			LM335			Units
		Min	Typ	Max	Min	Typ	Max	
Operating Output Voltage	$T_C = 25^\circ\text{C}$, $I_R = 1\text{ mA}$	2.95	2.98	3.01	2.92	2.98	3.04	V
Uncalibrated Temperature Error	$T_C = 25^\circ\text{C}$, $I_R = 1\text{ mA}$		1	3		2	6	°C
Uncalibrated Temperature Error	$T_{\text{MIN}} \leq T_C \leq T_{\text{MAX}}$, $I_R = 1\text{ mA}$		2	5		4	9	°C
Temperature Error with 25°C Calibration	$T_{\text{MIN}} \leq T_C \leq T_{\text{MAX}}$, $I_R = 1\text{ mA}$		0.5	1		1	2	°C
Calibrated Error at Extended Temperatures	$T_C = T_{\text{MAX}}$ (Intermittent)		2			2		°C
Non-Linearity	$I_R = 1\text{ mA}$		0.3	1.5		0.3	1.5	°C

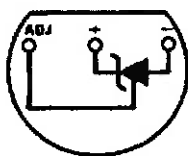
Electrical Characteristics (Note 1)

Parameter	Conditions	LM135/LM235 LM135A/LM235A			LM335 LM335A			Units
		Min	Typ	Max	Min	Typ	Max	
Operating Output Voltage	$400\text{ }\mu\text{A} \leq I_R \leq 5\text{ mA}$		2.5	10		3	14	mV
Change with Current	At Constant Temperature							
Dynamic Impedance	$I_R = 1\text{ mA}$		0.5			0.6		Ω
Output Voltage Temperature Coefficient			+10			+10		mV/°C
Time Constant	Still Air		80			80		sec
	100 ft./Min Air		10			10		sec
	Stirred Oil		1			1		sec
Time Stability	$T_C = 125^\circ\text{C}$		0.2			0.2		°C/yr



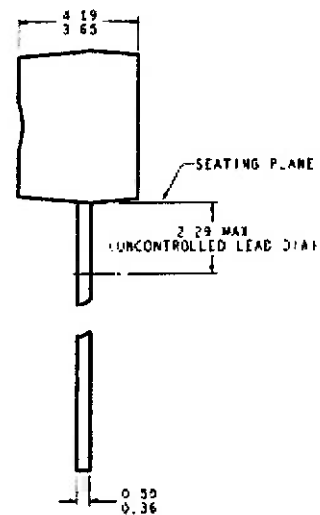
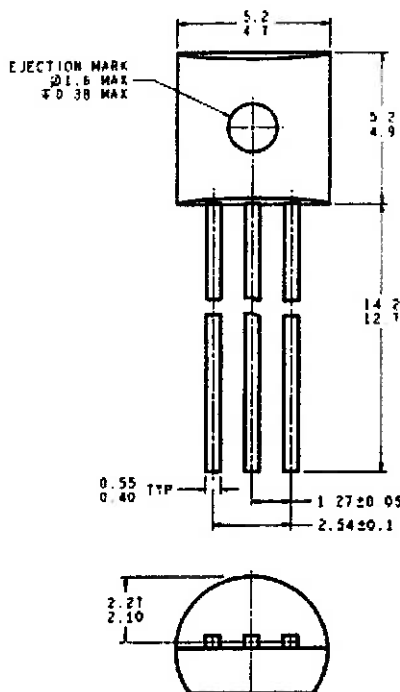


TO-92
Plastic Package



DS005688-5

Bottom View
Order Number LM335Z
or LM335AZ
See NS Package
Number Z03A



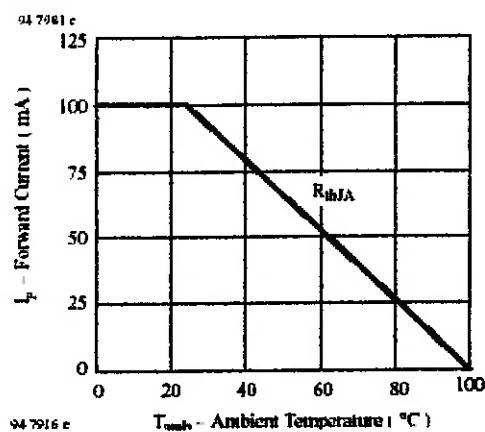
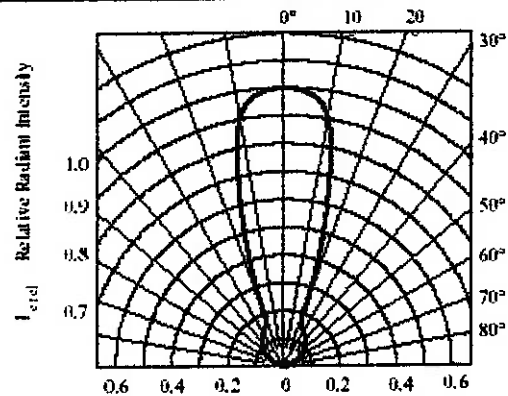
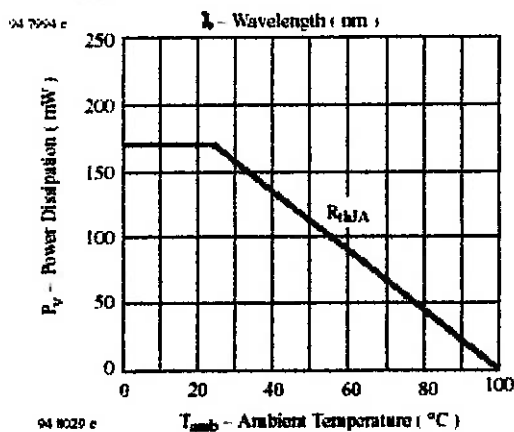
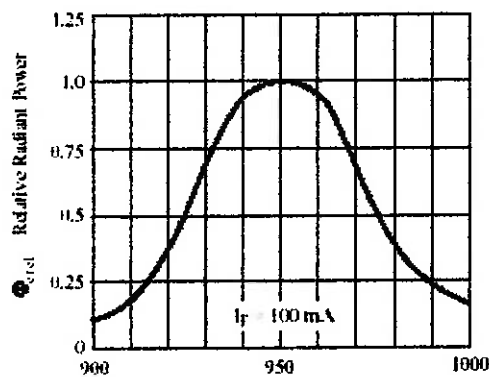
DIMENSIONS ARE IN MILLIMETERS

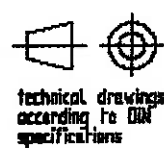
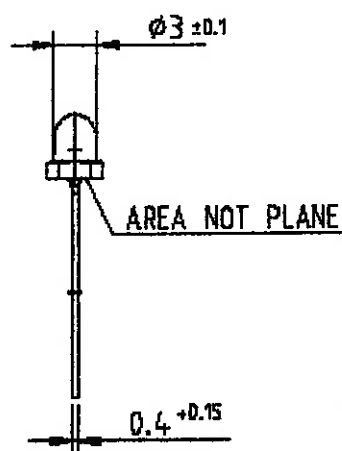
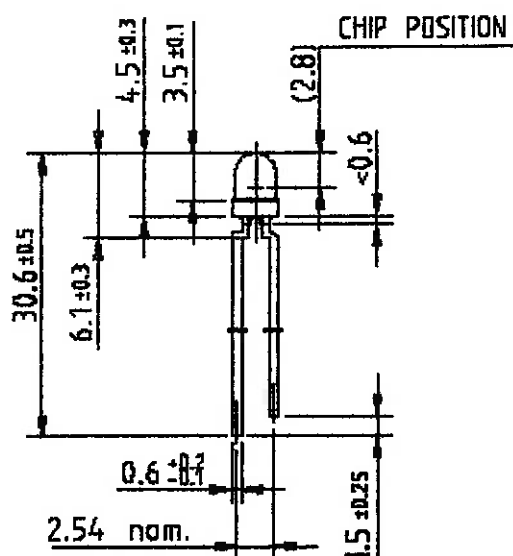
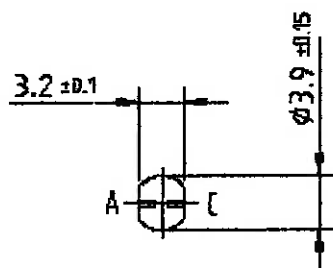
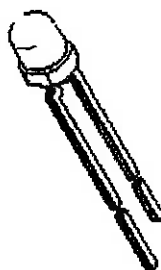
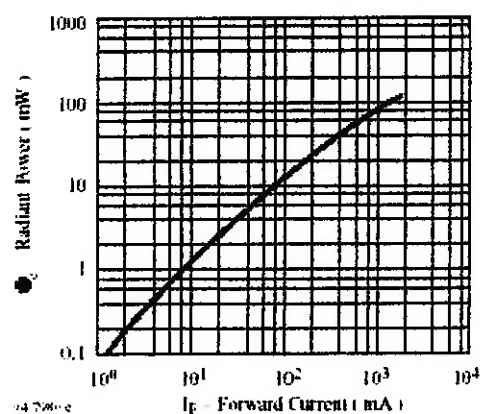
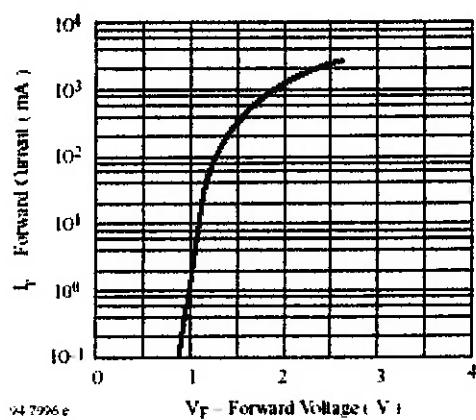
Plastic Package
Order Number LM335Z or LM335AZ
NS Package Z03A

A5. LED infravermelho TSUS4300 (TEMIC)

$T_{amb} = 25^{\circ}\text{C}$

Parameter	Test Conditions	Symbol	Min	Typ	Max	Unit
Forward Voltage	$I_F = 100 \text{ mA}, t_p = 20 \text{ ms}$	V_F		1.3	1.7	V
	$I_F = 1.5 \text{ A}, t_p = 100 \mu\text{s}$	V_F		2.2		V
Temp. Coefficient of V_F	$I_F = 100 \text{ mA}$	TK_{VF}		-1.3		mV/K
Reverse Current	$V_R = 5 \text{ V}$	I_R			100	μA
Breakdown Voltage	$I_R = 100 \mu\text{A}$	V_{BR}	5	40		
Junction Capacitance	$V_R = 0 \text{ V}, f = 1 \text{ MHz}, E = 0$	C_j		30		pF
Radiant Intensity	$I_F = 100 \text{ mA}, t_p = 20 \text{ ms}$	I_e	7	18		mW/sr
	$I_F = 1.5 \text{ A}, t_p = 100 \mu\text{s}$	I_e		160		mW/sr
Radiant Power	$I_F = 100 \text{ mA}, t_p = 20 \text{ ms}$	Φ_e		13		mW
Temp. Coefficient of Φ_e	$I_F = 20 \text{ mA}$	TK_{Φ_e}		-0.8		%/K
Angle of Half Intensity		φ		± 16		deg
Peak Wavelength	$I_F = 100 \text{ mA}$	λ_p		950		nm
Spectral Bandwidth	$I_F = 100 \text{ mA}$	$\Delta\lambda$		50		nm
Temp. Coefficient of λ_p	$I_F = 100 \text{ mA}$	TK_{λ_p}		0.2		nm/K
Rise Time	$I_F = 100 \text{ mA}$	t_r		800		ns
	$I_F = 1.5 \text{ A}$	t_r		400		ns
Fall Time	$I_F = 100 \text{ mA}$	t_f		800		ns
	$I_F = 1.5 \text{ A}$	t_f		400		ns





A6. Foto-diodo infravermelho BPV10NF (VISHAY)

Absolute Maximum Ratings

$T_{amb} = 25^{\circ}\text{C}$, unless otherwise specified

Parameter	Test condition	Symbol	Value	Unit
Reverse Voltage		V_R	60	V
Power Dissipation	$T_{amb} \leq 25^{\circ}\text{C}$	P_V	215	mW
Junction Temperature		T_j	100	$^{\circ}\text{C}$
Operating Temperature Range		T_{amb}	- 55 to + 100	$^{\circ}\text{C}$
Storage Temperature Range		T_{stg}	- 55 to + 100	$^{\circ}\text{C}$
Soldering Temperature	2 mm from body, $t \leq 5$ s	T_{sd}	260	$^{\circ}\text{C}$
Thermal Resistance Junction/ Ambient		R_{thJA}	350	K/W

Electrical Characteristics

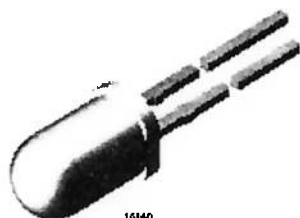
$T_{amb} = 25^{\circ}\text{C}$, unless otherwise specified

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Forward Voltage	$I_F = 50$ mA	V_F		1	1.3	V
Breakdown Voltage	$I_R = 100$ μA , $E = 0$	$V_{(BR)}$	60			V
Reverse Dark Current	$V_R = 20$ V, $E = 0$	I_{n0}		1	5	nA
Diode capacitance	$V_R = 0$ V, $f = 1$ MHz, $E = 0$	C_D		11		pF

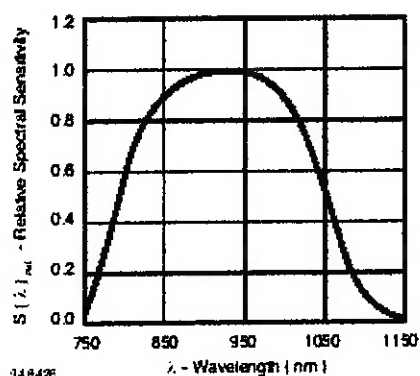
Optical Characteristics

$T_{amb} = 25^{\circ}\text{C}$, unless otherwise specified

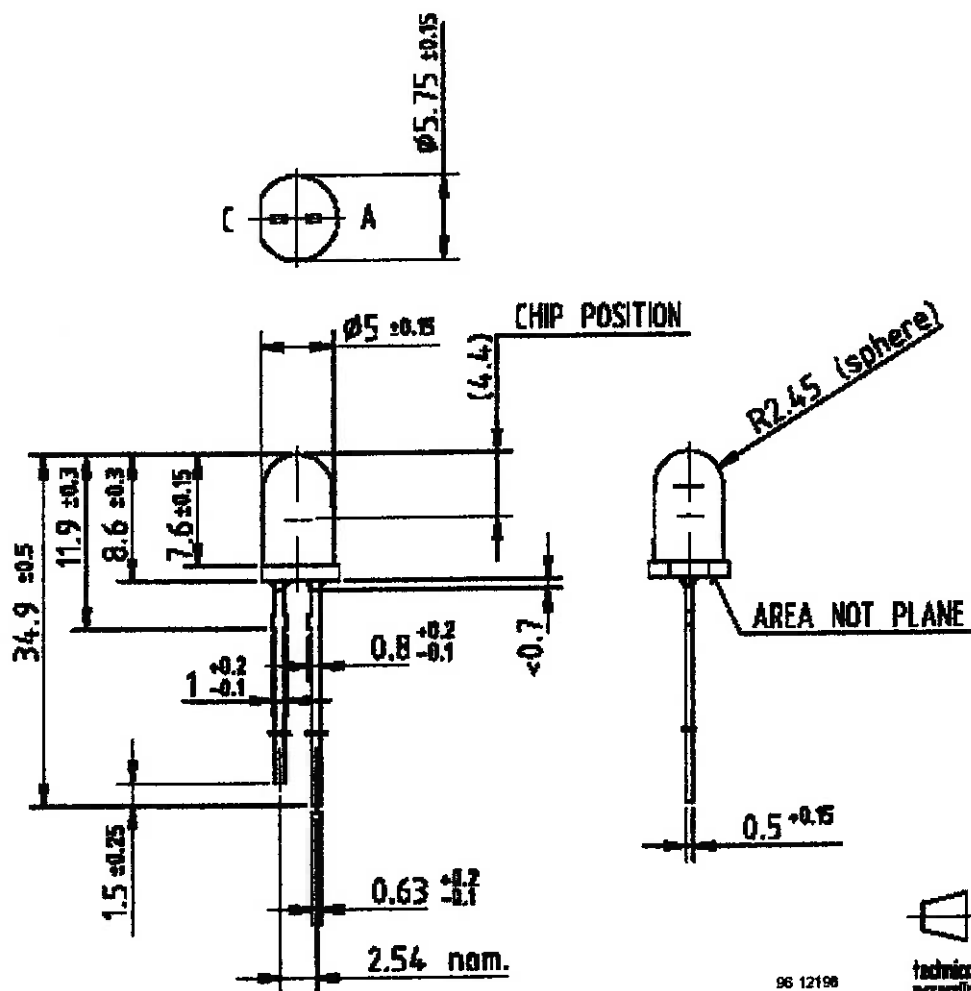
Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Open Circuit Voltage	$E_e = 1$ mW/cm ² , $\lambda = 870$ nm	V_o		450		mV
Short Circuit Current	$E_e = 1$ mW/cm ² , $\lambda = 870$ nm	I_k		50		μA
Reverse Light Current	$E_e = 1$ mW/cm ² , $\lambda = 870$ nm, $V_R = 5$ V	I_{ra}		55		μA
	$E_e = 1$ mW/cm ² , $\lambda = 950$ nm, $V_R = 5$ V	I_{ra}	30	60		μA
Temp. Coefficient of I_{ra}	$E_e = 1$ mW/cm ² , $\lambda = 870$ nm, $V_R = 5$ V	TK_{Ra}		-0.1		%/K
Absolute Spectral Sensitivity	$V_R = 5$ V, $\lambda = 870$ nm	$s(\lambda)$		0.55		A/W
Angle of Half Sensitivity		ϕ		± 20		deg
Wavelength of Peak Sensitivity		λ_p		940		nm
Range of Spectral Bandwidth		$\lambda_{0.6}$		790 to 1050		nm
Quantum Efficiency	$\lambda = 950$ nm	η		70		%
Noise Equivalent Power	$V_R = 20$ V, $\lambda = 950$ nm	NEP		3×10^{-14}		W/ $\sqrt{\text{Hz}}$
Detectivity	$V_R = 20$ V, $\lambda = 950$ nm	D^*		3×10^{12}		cm $\sqrt{\text{Hz}}$ /W
Rise Time	$V_R = 50$ V, $R_L = 50$ Ω , $\lambda = 820$ nm	t_r		2.5		ns
Fall Time	$V_R = 50$ V, $R_L = 50$ Ω , $\lambda = 820$ nm	t_f		2.5		ns



16140



14842E



98 12198

technical drawings
according to DIN
specifications

A7. Sensor de Ângulo de Ataque e Sideslip

ANEXO B – Circuitos Integrados

B1. Bus Transceiver com tri-state 74LS245N (Texas Instruments)

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITION†	SN54LS245			SN74LS245			UNIT
				MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V _{IH}	High-level input voltage			2			2			V
V _{IL}	Low-level input voltage			0.7			0.8			V
V _{IK}	Input clamp voltage		V _{CC} = MIN. I _I = -18 mA	-1.5			-1.5			V
Hysteresis (V _{T+} - V _{T-})		A or B	V _{CC} = MIN.	0.2	0.4		0.2	0.4		V
V _{OH}	High-level output voltage	V _{CC} = MIN. V _{IH} = 2 V. V _{IL} = V _{IL(max)}	I _{OH} = -3 mA	2.4	3.4		2.4	3.4		V
			I _{OH} = MAX	2			2			
V _{OL}	Low-level output voltage	V _{CC} = MIN. V _{IH} = 2 V. V _{IL} = V _{IL(max)}	I _{OL} = 12 mA	0.4			0.4		V	
			I _{OL} = 24 mA			0.5				
I _{OZH}	Off-state output current, high-level voltage applied		V _{CC} = MAX. OE at 2 V V _O = 2.7 V	20			20		μA	
I _{OZL}	Off-state output current, low-level voltage applied		V _{CC} = MAX. OE at 2 V V _O = 0.4 V	-200			-200		μA	
I _I	Input current at maximum input voltage	A or B	V _{CC} = MAX	V _I = 5.5 V	0.1		0.1		mA	
		DIR or OE		V _I = 7 V	0.1		0.1			
I _{IH}	High-level input current		V _{CC} = MAX. V _{IH} = 2.7 V	20			20		μA	
I _{IL}	Low-level input current		V _{CC} = MAX. V _{IL} = 0.4 V	-0.2			-0.2		mA	
I _{OS}	Short-circuit output current§		V _{CC} = MAX	-40	-225		40	-225	mA	
I _{CC}	Supply current	Total, outputs high	V _{CC} = MAX	Outputs open	48	70		48	70	mA
		Total, outputs low			62	90		62	90	
		Outputs at high Z			64	95		64	95	

† For conditions shown as MIN or MAX, use the appropriate values specified under recommended operating conditions.

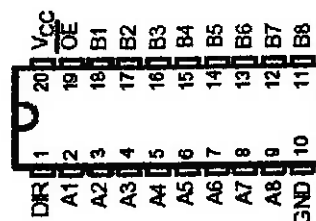
‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time, and duration of the short circuit should not exceed one second.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$ (see Figure 1)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low- to high-level output		8	12	ns
t_{PHL}	Propagation delay time, high- to low-level output		8	12	
t_{PZL}	Output enable time to low level		27	40	ns
t_{PZH}	Output enable time to high level		25	40	
t_{PLZ}	Output disable time from low level		15	25	ns
t_{PHZ}	Output disable time from high level		15	28	

FUNCTION TABLE		
INPUTS		OPERATION
OE	DIR	
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

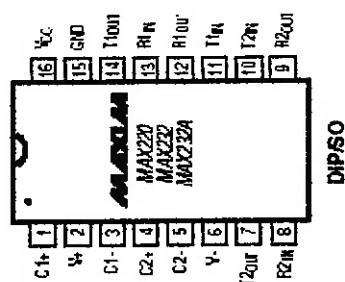


B2. Conversor de Nível HIN232CP (similar ao MAX232 – MAXIM)

ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241 (continued)

(MAX223/230/232/234/236/237/238/240/241, $V_{CC} = +5V \pm 10\%$; MAX233/MAX235, $V_{CC} = 5V \pm 5\%$, C1–C4 = 1.0 μ F; MAX231/MAX239, $V_{CC} = 5V \pm 10\%$; $V_+ = 7.5V$ to 13.2V; $T_A = T_{MIN}$ to T_{MAX} ; unless otherwise noted.)

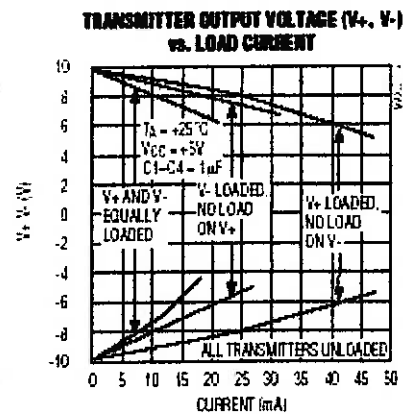
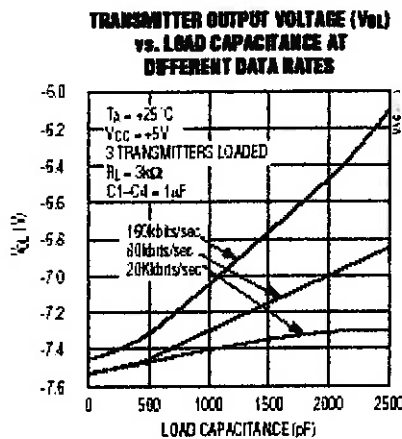
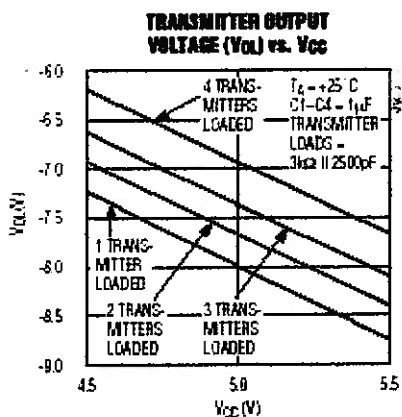
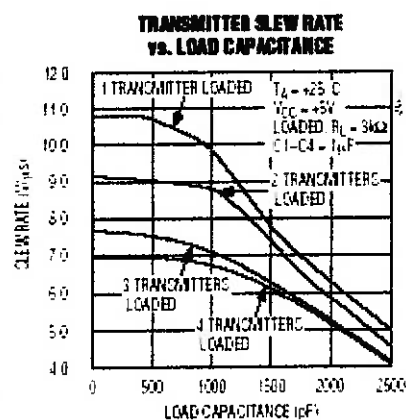
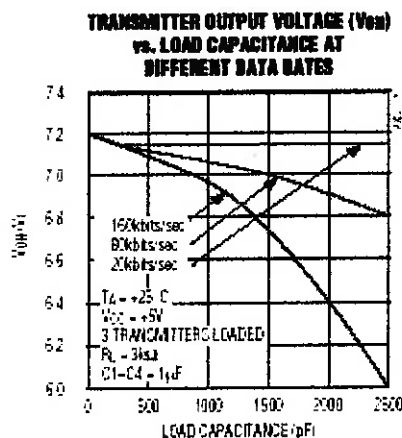
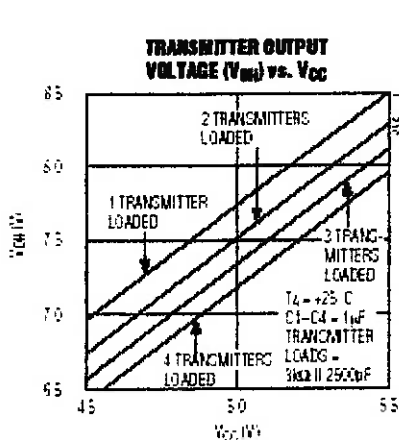
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 Input Threshold Low	$T_A = +25^{\circ}\text{C}$, $V_{CC} = 5\text{V}$	Normal operation S1DN = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)	0.8	1.2		V
		Shutdown (MAX223) SHDN = 0V, EN = 5V (R4IN, R5IN)	0.6	1.5		
RS-232 Input Threshold High	$T_A = +25^{\circ}\text{C}$, $V_{CC} = 5\text{V}$	Normal operation SHDN = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)		1.7	2.4	V
		Shutdown (MAX223) SHDN = 0V, EN = 5V (R4IN, R5IN)		1.5	2.4	
RS-232 Input Hysteresis	$V_{CC} = 5\text{V}$, no hysteresis in shutdown		0.2	0.5	1.0	V
RS-232 Input Resistance	$T_A = +25^{\circ}\text{C}$, $V_{CC} = 5\text{V}$		3	5	7	k Ω
TTL/CMOS Output Voltage Low	$I_{OUT} = 1.6\text{mA}$ (MAX231/232/233, $I_{OUT} = 3.2\text{mA}$)				0.4	V
TTL/CMOS Output Voltage High	$I_{OUT} = -1\text{mA}$		3.5	$V_{CC} - 0.4$		V
TTL/CMOS Output Leakage Current	$0\text{V} \leq R_{OUT} \leq V_{CC}$; EN = 0V (MAX223); EN = V_{CC} (MAX235–241)			0.05	± 10	μA
Receiver Output Enable Time	Normal operation	MAX223		600		ns
		MAX235/236/239/240/241		400		
Receiver Output Disable Time	Normal operation	MAX223		300		ns
		MAX235/236/239/240/241		250		
Propagation Delay	RS-232 IN to TTL/CMOS OUT, $C_L = 150\text{pF}$	Normal operation		0.5	10	μs
		SHDN = 0V (MAX223)	tPHLS	4	40	
			tPLHS	6	40	
Transition Region Slew Rate	MAX223/MAX230/MAX234–241, $T_A = +25^{\circ}\text{C}$, $V_{CC} = 5\text{V}$, $R_L = 3\text{k}\Omega$ to 7k Ω , $C_L = 50\text{pF}$ to 2500pF, measured from +3V to -3V or -3V to +3V		3	5.1	30	V/ μs
	MAX231/MAX232/MAX233, $T_A = +25^{\circ}\text{C}$, $V_{CC} = 5\text{V}$, $R_L = 3\text{k}\Omega$ to 7k Ω , $C_L = 50\text{pF}$ to 2500pF, measured from +3V to -3V or -3V to +3V			4	30	
Transmitter Output Resistance	$V_{CC} = V_+ = V_- = 0\text{V}$, $V_{OUT} = \pm 2\text{V}$		300			Ω
Transmitter Output Short-Circuit Current				± 10		mA



ELECTRICAL CHARACTERISTICS—MAX223/MAX230—MAX241

(MAX223/230/232/234/236/237/238/240/241, $V_{CC} = +5V \pm 10\%$; MAX233/MAX235, $V_{CC} = 5V \pm 5\%$, $C1-C4 = 1.0\mu F$; MAX231/MAX239, $V_{CC} = 5V \pm 10\%$; $V_+ = 7.5V$ to $13.2V$; $T_A = T_{MIN}$ to T_{MAX} ; unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Output Voltage Swing	All transmitter outputs loaded with $3k\Omega$ to ground	± 5.0	± 7.3		V
V_{CC} Power-Supply Current	No load, $T_A = +25^\circ C$	MAX232/233		5	10
		MAX223/230/234-238/240/241		7	15
		MAX231/239		0.4	1
V_+ Power-Supply Current		MAX231		1.8	5
		MAX239		5	15
Shutdown Supply Current	$T_A = +25^\circ C$	MAX223		15	50
		MAX230/235/236/240/241		1	10
Input Logic Threshold Low	T_{IL} ; EN, SHDN (MAX233); EN, SHDN (MAX230/235-241)			0.8	V
Input Logic Threshold High	T_{IH}		2.0		V
	EN, SHDN (MAX233); EN, SHDN (MAX230/235/236/240/241)		2.4		V
Logic Pull-Up Current	$T_{IH} = 0V$		1.5	200	μA
Receiver Input Voltage Operating Range		-30		30	V

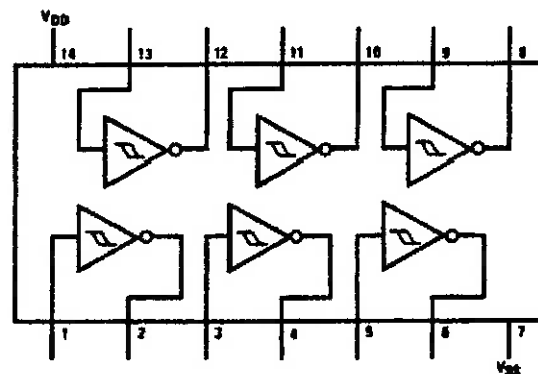


B3. Inversor CD40106BC (National Semiconductor)

DC Electrical Characteristics CD40106BC (Note 2)

Symbol	Parameter	Conditions	-40°C		+25°C			+85°C		Units
			Min	Max	Min	Typ	Max	Min	Max	
I_{DD}	Quiescent Device Current	$V_{DD} = 5V$		4.0			4.0		30	μA
		$V_{DD} = 10V$		8.0			8.0		60	μA
		$V_{DD} = 15V$		16.0			16.0		120	μA
V_{OL}	Low Level Output Voltage	$ I_O < 1 \mu A$								
		$V_{DD} = 5V$		0.05			0.05		0.05	V
		$V_{DD} = 10V$		0.05			0.05		0.05	V
V_{OH}	High Level Output Voltage	$ I_O < 1 \mu A$								
		$V_{DD} = 5V$	4.95		4.95	5		4.95		V
		$V_{DD} = 10V$	9.95		9.95	10		9.95		V
V_{T-}	Negative-Going Threshold Voltage	$V_{DD} = 5V, V_O = 4.5V$	0.7	2.0	0.7	1.4	2.0	0.7	2.0	V
		$V_{DD} = 10V, V_O = 9V$	1.4	4.0	1.4	3.2	4.0	1.4	4.0	V
		$V_{DD} = 15V, V_O = 13.5V$	2.1	6.0	2.1	5.0	6.0	2.1	6.0	V
V_{T+}	Positive-Going Threshold Voltage	$V_{DD} = 5V, V_O = 0.5V$	3.0	4.3	3.0	3.6	4.3	3.0	4.3	V
		$V_{DD} = 10V, V_O = 1V$	6.0	8.6	6.0	6.8	8.6	6.0	8.6	V
		$V_{DD} = 15V, V_O = 1.5V$	9.0	12.9	9.0	10.0	12.9	9.0	12.9	V
V_H	Hysteresis ($V_{T+} - V_{T-}$) Voltage	$V_{DD} = 5V$	1.0	3.6	1.0	2.2	3.6	1.0	3.6	V
		$V_{DD} = 10V$	2.0	7.2	2.0	3.6	7.2	2.0	7.2	V
		$V_{DD} = 15V$	3.0	10.8	3.0	5.0	10.8	3.0	10.8	V
I_{OL}	Low Level Output Current (Note 3)	$V_{DD} = 5V, V_O = 0.4V$	0.52		0.44	0.88		0.36		mA
		$V_{DD} = 10V, V_O = 0.5V$	1.3		1.1	2.25		0.9		mA
		$V_{DD} = 15V, V_O = 1.5V$	3.6		3.0	8.8		2.4		mA
I_{OH}	High Level Output Current (Note 3)	$V_{DD} = 5V, V_O = 4.6V$	-0.52		-0.44	0.88		-0.36		mA
		$V_{DD} = 10V, V_O = 9.5V$	-1.3		-1.1	2.25		-0.9		mA
		$V_{DD} = 15V, V_O = 13.5V$	-3.6		-3.0	8.8		-2.4		mA
I_{IN}	Input Current	$V_{DD} = 15V, V_{IN} = 0V$		0.30		10^{-5}	0.30		-1.0	μA
		$V_{DD} = 15V, V_{IN} = 15V$		0.30		10^{-5}	0.30		1.0	μA

Dual-In-Line Package

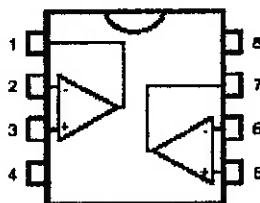


B4. Amplificador Operacional LM358N (ST Microelectronics)

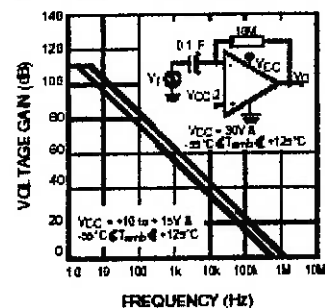
ELECTRICAL CHARACTERISTICS

$V_{CC}^+ = +5V$, $V_{CC}^- = \text{Ground}$, $V_O = 1.4V$, $T_{amb} = +25^\circ C$ (unless otherwise specified)

Symbol	Parameter	LM158A-LM258A LM358A			LM158-LM258 LM358			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
V_{IO}	Input Offset Voltage - note 1) $T_{amb} = +25^\circ C$ LM158, LM258 LM158A $T_{min} \leq T_{amb} \leq T_{max}$ LM158, LM258		1	3 2 4		2	7 5 9 7	mV
I_{IO}	Input Offset Current $T_{amb} = +25^\circ C$ $T_{min} \leq T_{amb} \leq T_{max}$		2	10 30		2	30 40	nA
I_{IB}	Input Bias Current - note 2) $T_{amb} = +25^\circ C$ $T_{min} \leq T_{amb} \leq T_{max}$		20	50 100		20	150 200	nA
A_{vd}	Large Signal Voltage Gain $V_{CC} = +15V$, $R_L = 2k\Omega$, $V_O = 1.4V$ to $11.4V$ $T_{amb} = +25^\circ C$ $T_{min} \leq T_{amb} \leq T_{max}$	50 25	100		50 25	100		V/mV
SVR	Supply Voltage Rejection Ratio ($R_S \leq 10k\Omega$) $V_{CC}^+ = 5V$ to $30V$ $T_{amb} = +25^\circ C$ $T_{min} \leq T_{amb} \leq T_{max}$	65 65	100		65 65	100		dB
I_{CC}	Supply Current, all Amp. no load $T_{min} \leq T_{amb} \leq T_{max}$ $T_{min} \leq T_{amb} \leq T_{max}$ $V_{CC} = +5V$ $V_{CC} = +30V$		0.7	1.2 1		0.7	1.2 2	mA
V_{ICM}	Input Common Mode Voltage Range $V_{CC} = +30V$ - note 3) $T_{amb} = +25^\circ C$ $T_{min} \leq T_{amb} \leq T_{max}$	0 0		$V_{CC}^+ - 1.5$ $V_{CC}^+ - 2$	0 0		$V_{CC}^+ - 1.5$ $V_{CC}^+ - 2$	V
CMR	Common Mode Rejection Ratio ($R_S \leq 10k\Omega$) $T_{amb} = +25^\circ C$ $T_{min} \leq T_{amb} \leq T_{max}$	70 60	85		70 60	85		dB
I_{SOURCE}	Output Current Source $V_{CC} = +15V$, $V_O = +2V$, $V_{ID} = +1V$	20	40	60	20	40	60	mA
I_{SINK}	Output Sink Current ($V_{ID} = -1V$) $V_{CC} = +15V$, $V_O = +2V$ $V_{CC} = +15V$, $V_O = +0.2V$	10 12	20 50		10 12	20 50		mA μA
V_{OPP}	Output Voltage Swing ($R_L = 2k\Omega$) $T_{amb} = +25^\circ C$ $T_{min} \leq T_{amb} \leq T_{max}$	0 0		$V_{CC}^+ - 1.5$ $V_{CC}^+ - 2$	0 0		$V_{CC}^+ - 1.5$ $V_{CC}^+ - 2$	



- 1 - Output 1
- 2 - Inverting input
- 3 - Non-inverting input
- 4 - V_{CC}^-
- 5 - Non-inverting input 2
- 6 - Inverting input 2
- 7 - Output 2
- 8 - V_{CC}^+



Symbol	Parameter	LM158A-LM258A LM358A			LM158-LM258 LM358			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
V_{OH}	High Level Output Voltage ($V_{CC}^+ = 30V$) $T_{amb} = +25^\circ C$ $R_L = 2k\Omega$ $T_{min} \leq T_{amb} \leq T_{max}$ $T_{amb} = +25^\circ C$ $R_L = 10k\Omega$ $T_{min} \leq T_{amb} \leq T_{max}$	26 26 27 27	27 28		26 26 27 27	27 28		V
V_{OL}	Low Level Output Voltage ($R_L = 10k\Omega$) $T_{amb} = +25^\circ C$ $T_{min} \leq T_{amb} \leq T_{max}$		5	20 20		5 20 20		mV
SR	Slew Rate $V_{CC} = 15V$, $V_i = 0.5$ to $3V$, $R_L = 2k\Omega$, $C_L = 100pF$, unity Gain	0.3	0.6		0.3	0.6		V/ μs
GBP	Gain Bandwidth Product $V_{CC} = 30V$, $f = 100kHz$, $V_{in} = 10mV$, $R_L = 2k\Omega$, $C_L = 100pF$	0.7	1.1		0.7	1.1		MHz
THD	Total Harmonic Distortion $f = 1kHz$, $A_v = 20dB$, $R_L = 2k\Omega$, $V_O = 2V_{pp}$, $C_L = 100pF$, $V_O = 2V_{pp}$		0.02			0.02		%
e_n	Equivalent Input Noise Voltage $f = 1kHz$, $R_s = 100\Omega$, $V_{CC} = 30V$		55			55		$\frac{nV}{\sqrt{Hz}}$
DV_{IO}	Input Offset Voltage Drift		7	15		7	30	$\mu V/^\circ C$
DI_{IO}	Input Offset Current Drift		10	200		10	300	$pA/^\circ C$
V_{OI}/V_{O2}	Channel Separation - note 4) $1kHz \leq f \leq 20kHz$		120			120		dB

- $V_O = 1.4V$, $R_s = 0\Omega$, $5V < V_{CC}^+ < 30V$, $0 < V_{IC} < V_{CC}^+ - 1.5V$
- The direction of the input current is out of the IC. This current is essentially constant. Independent of the state of the output so no loading change exists on the input lines.
- The input common-mode voltage of either input signal voltage should not be allowed to go negative by more than $0.3V$. The upper end of the common-mode voltage range is $V_{CC}^+ - 1.5V$, but either or both inputs can go to $+32V$ without damage.
- Due to the proximity of external components insure that coupling is not originating via stray capacitance between these external parts. This typically can be relieved at this type of parasitics increases at higher frequencies.

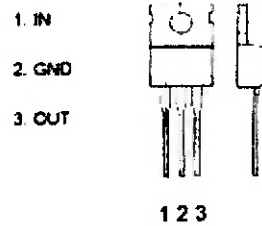
ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	LM158A	LM258A	LM358A	Unit
V_{CC}	Supply voltage		+32		V
V_i	Input Voltage		-0.3 to +32		V
V_{id}	Differential Input Voltage		+32		V
P_{tot}	Power Dissipation ¹⁾		500		mW
	Output Short-circuit Duration ²⁾		Infinite		
I_{in}	Input Current ³⁾		50		mA
T_{oper}	Operating Free-air Temperature Range	-55 to +125	-40 to +105	0 to +70	$^\circ C$
T_{stg}	Storage Temperature Range		-65 to +150		$^\circ C$

- Power dissipation must be considered to ensure maximum junction temperature (T_J) is not exceeded.
- Short-circuits from the output to V_{CC} can cause excessive heating if $V_{CC} > 15V$. The maximum output current is approximately $40mA$ independent of the magnitude of V_{CC} . Destructive dissipation can result from simultaneous short-circuit on all amplifiers.
- This input current only exists when the voltage at any of the input leads is driven negative. It is due to the collector-base junction of the input PNP transistor becoming forward biased and thereby acting as input diodes clamps. In addition to this diode action, there is also NPN parasitic action on the IC chip. This transistor action can cause the output voltages of the Op-amps to go to the V_{CC} voltage level (or to ground for a large overdrive) for the time duration that an input is driven negative. This is not destructive and normal output will set up again for input voltage higher than $-0.3V$.

B5. Regulador 5V LM7805 (HsiN Semiconductor)

TO-220



FEATURES

Maximum Output current

I_{OM} 1.5 A

Output voltage

V_o 5 V

ABSOLUTE MAXIMUM RATINGS (Operating temperature range applies unless otherwise specified)

Parameter	Symbol	Value	Unit
Input Voltage	V_I	35	V
Operating Junction Temperature Range	T_{OPR}	0~+125	°C
Storage Temperature Range	T_{STG}	-55~+150	°C

ELECTRICAL CHARACTERISTICS ($V_I=10V, I_o=500mA, 0^\circ C < T_J < 125^\circ C, C_I=0.33\mu F, C_o=0.1\mu F$, unless otherwise specified)

Parameter	Symbol	Test conditions	MIN	TYP	MAX	UNIT
Output voltage	V_o	$T_J=25^\circ C$	4.8	5.0	5.2	V
		$7V \leq V_I \leq 20V, I_o=5mA \sim 1A, P_o < 15W$	4.75	5.00	5.25	V
Load Regulation	ΔV_o	$T_J=25^\circ C, I_o=5mA \sim 1.5A$		9	100	mV
		$T_J=25^\circ C, I_o=250mA \sim 750mA$		4	50	mV
Line regulation	ΔV_o	$7V \leq V_I \leq 25V, T_J=25^\circ C$		4	100	mV
		$8V \leq V_I \leq 12V, T_J=25^\circ C$		1.6	50	mV
Quiescent Current	I_q	$T_J=25^\circ C$		5	8	mA
Quiescent Current Change	ΔI_q	$7V \leq V_I \leq 25V$		0.3	1.3	mA
	ΔI_q	$5mA \leq I_o \leq 1A$		0.03	0.5	mA
Output Noise Voltage	V_n	10Hz $\leq f \leq$ 100KHz		42		μV
Ripple Rejection	RR	$8V \leq V_I \leq 18V, f=120Hz, T_J=25^\circ C$	62	73		dB
Dropout Voltage	V_d	$T_J=25^\circ C, I_o=1A$		2		V
Short Circuit Current	I_{sc}	$V_I=35V, T_a=25^\circ C$		230		mA
Peak Current	I_{pk}	$T_J=25^\circ C$		2.2		A

B6. Regulador 3,3V LM1117DT

Electrical Characteristics

Typicals and limits appearing in normal type apply for $T_J = 25^\circ\text{C}$. Limits appearing in Boldface type apply over the entire junction temperature range for operation, 0°C to 125°C .

Symbol	Parameter	Conditions	Min (Note 5)	Typ (Note 4)	Max (Note 5)	Units
V_{REF}	Reference Voltage	LM1117-ADJ $I_{\text{OUT}}=10\text{mA}$, $V_{\text{IN}}-V_{\text{OUT}}=2\text{V}$, $T_J=25^\circ\text{C}$ $10\text{mA} \leq I_{\text{OUT}} \leq 800\text{mA}$, $1.4\text{V} \leq V_{\text{IN}}-V_{\text{OUT}} \leq 10\text{V}$	1.238 1.225	1.250 1.250	1.262 1.270	V V
		LM1117-2.85 $I_{\text{OUT}}=10\text{mA}$, $V_{\text{IN}}=4.85\text{V}$, $T_J=25^\circ\text{C}$ $0 \leq I_{\text{OUT}} \leq 800\text{mA}$, $4.25\text{V} \leq V_{\text{IN}} \leq 10\text{V}$ $0 \leq I_{\text{OUT}} \leq 500\text{mA}$, $V_{\text{IN}}=4.10\text{V}$	2.820 2.790 2.790	2.850 2.850 2.850	2.880 2.910 2.910	V V V
V_{OUT}	Output Voltage	LM1117-3.3 $I_{\text{OUT}}=10\text{mA}$, $V_{\text{IN}}=5\text{V}$, $T_J=25^\circ\text{C}$ $0 \leq I_{\text{OUT}} \leq 800\text{mA}$, $4.75\text{V} \leq V_{\text{IN}} \leq 10\text{V}$	3.267 3.235	3.300 3.300	3.333 3.365	V V
		LM1117-5.0 $I_{\text{OUT}}=10\text{mA}$, $V_{\text{IN}}=7\text{V}$, $T_J=25^\circ\text{C}$ $0 \leq I_{\text{OUT}} \leq 800\text{mA}$, $6.5\text{V} \leq V_{\text{IN}} \leq 12\text{V}$	4.950 4.900	5.000 5.000	5.050 5.100	V V
		LM1117-ADJ $I_{\text{OUT}}=10\text{mA}$, $1.5\text{V} \leq V_{\text{IN}}-V_{\text{OUT}} \leq 13.75\text{V}$		0.035	0.2	%
		LM1117-2.85 $I_{\text{OUT}}=0\text{mA}$, $4.25\text{V} \leq V_{\text{IN}} \leq 10\text{V}$		1	6	mV
ΔV_{OUT}	Line Regulation (Note 6)	LM1117-3.3 $I_{\text{OUT}}=0\text{mA}$, $4.75\text{V} \leq V_{\text{IN}} \leq 15\text{V}$		1	6	mV
		LM1117-5.0 $I_{\text{OUT}}=0\text{mA}$, $6.5\text{V} \leq V_{\text{IN}} \leq 15\text{V}$		1	10	mV
		LM1117-ADJ $V_{\text{IN}}-V_{\text{OUT}}=3\text{V}$, $10 \leq I_{\text{OUT}} \leq 800\text{mA}$		0.2	0.4	%
		LM1117-2.85 $V_{\text{IN}}=4.25\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	10	mV
ΔV_{OUT}	Load Regulation (Note 6)	LM1117-3.3 $V_{\text{IN}}=4.75\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	10	mV
		LM1117-5.0 $V_{\text{IN}}=6.5\text{V}$, $0 \leq I_{\text{OUT}} \leq 800\text{mA}$		1	15	mV
		$I_{\text{OUT}}=100\text{mA}$		1.10	1.20	V
		$I_{\text{OUT}}=500\text{mA}$		1.15	1.25	V
$V_{\text{IN}}-V_{\text{OUT}}$	Dropout Voltage (Note 7)	$I_{\text{OUT}}=800\text{mA}$		1.20	1.30	V
		$V_{\text{IN}}-V_{\text{OUT}}=5\text{V}$, $T_J=25^\circ\text{C}$	800	1200	1500	mA
		LM1117-ADJ $V_{\text{IN}}=15\text{V}$		1.7	5	mA

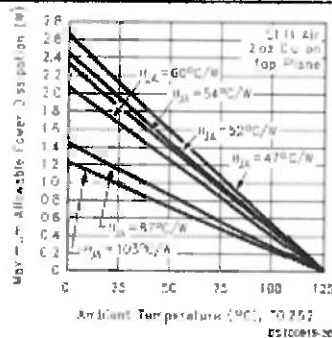
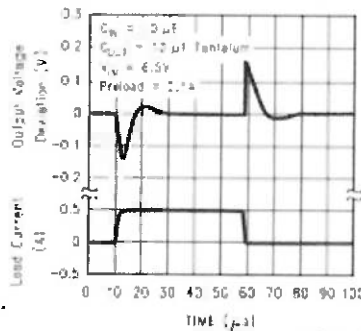
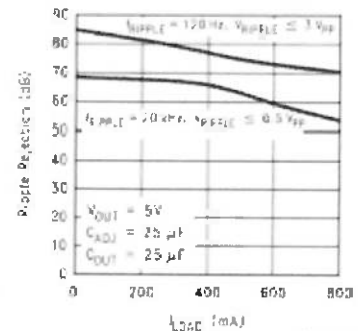


FIGURE 10. Maximum Allowable Power Dissipation vs. Ambient Temperature for TO-252

LM1117-5.0 Load Transient Response



LM1117-ADJ Ripple Rejection vs. Current



Electrical Characteristics (Continued)

Typicals and limits appearing in normal type apply for $T_J = 25^\circ\text{C}$. Limits appearing in Boldface type apply over the entire junction temperature range for operation, 0°C to 125°C .

Symbol	Parameter	Conditions	Min (Note 5)	Typ (Note 4)	Max (Note 5)	Units
	Quiescent Current	LM1117-2.85 $V_{IN} \leq 10\text{V}$		5	10	mA
		LM1117-3.3 $V_{IN} \leq 15\text{V}$		5	10	mA
		LM1117-5.0 $V_{IN} \leq 15\text{V}$		5	10	mA
	Thermal Regulation	$T_A = 25^\circ\text{C}$, 30ms Pulse		0.01	0.1	%/W
	Ripple Regulation	$f_{\text{RIPPLE}} = 120\text{Hz}$, $V_{IN} - V_{OUT} = 3\text{V}$ $V_{\text{RIPPLE}} = 1\text{V}_{\text{PP}}$	60	75		dB
	Adjust Pin Current			60	120	μA
	Adjust Pin Current Change	$10 \leq I_{OUT} \leq 800\text{mA}$, $1.4\text{V} \leq V_{IN} - V_{OUT} \leq 10\text{V}$		0.2	5	μA
	Temperature Stability			0.5		%
	Long Term Stability	$T_A = 125^\circ\text{C}$, 1000Hrs		0.3		%
	RMS Output Noise	(% of V_{OUT}), $10\text{Hz} \leq f \leq 10\text{kHz}$		0.003		%
	Thermal Resistance Junction-to-Case	3-Lead SOT-223		15.0		$^\circ\text{C/W}$
		3-Lead TO-220		3.0		$^\circ\text{C/W}$
		3-Lead TO-252		10		$^\circ\text{C/W}$
	Thermal Resistance Junction-to-Ambient (No heat sink; No air flow)	3-Lead SOT-223		136		$^\circ\text{C/W}$
		3-Lead TO-220		79		$^\circ\text{C/W}$
		3-Lead TO-252 (Note 9)		92		$^\circ\text{C/W}$

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but specific performance is not guaranteed. For guaranteed specifications and the test conditions, see the Electrical Characteristics.

Note 2: The maximum power dissipation is a function of $T_{J(\text{max})}$, θ_{JA} , and T_A . The maximum allowable power dissipation at any ambient temperature is $P_D = (T_J(\text{max}) - T_A) / \theta_{JA}$. All numbers apply for packages soldered directly into a PC board.

Note 3: For testing purposes, ESD was applied using human body model, $1.5\text{k}\Omega$ in series with 100pF .

Note 4: Typical Values represent the most likely parametric norm.

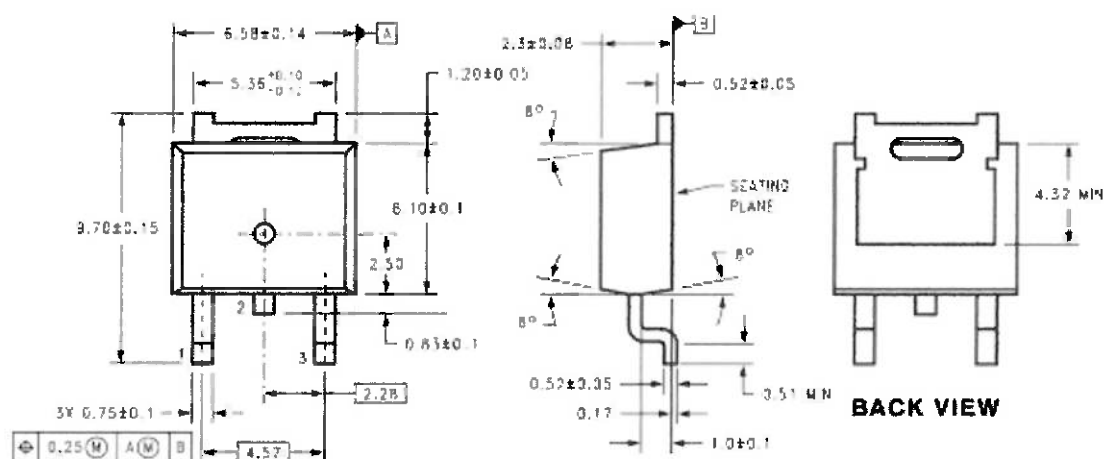
Note 5: All limits are guaranteed by testing or statistical analysis.

Note 6: Load and line regulation are measured at constant junction room temperature.

Note 7: The dropout voltage is the input/output differential at which the circuit ceases to regulate against further reduction in input voltage. It is measured when the output voltage has dropped 100mV from the nominal value obtained at $V_{IN} = V_{OUT} + 1.5\text{V}$.

Note 8: The minimum output current required to maintain regulation.

Note 9: Minimum pad size of 0.038in^2 .



DIMENSIONS ARE IN MILLIMETERS

3-Lead TO-252 Package

Order Number LM1117DT-ADJ, LM1117DT-2.85, LM1117DT-3.3, or LM1117DT-5.0

NSC Package Number TD03B

©2000 NSC

ANEXO C – Multimedia Card SDMJ-128 SanDisk e Memória 24LC04B Microchip

C1. Multimedia Card SDMJ-128 SanDisk

Figure 1-2 Memory Array Partitioning

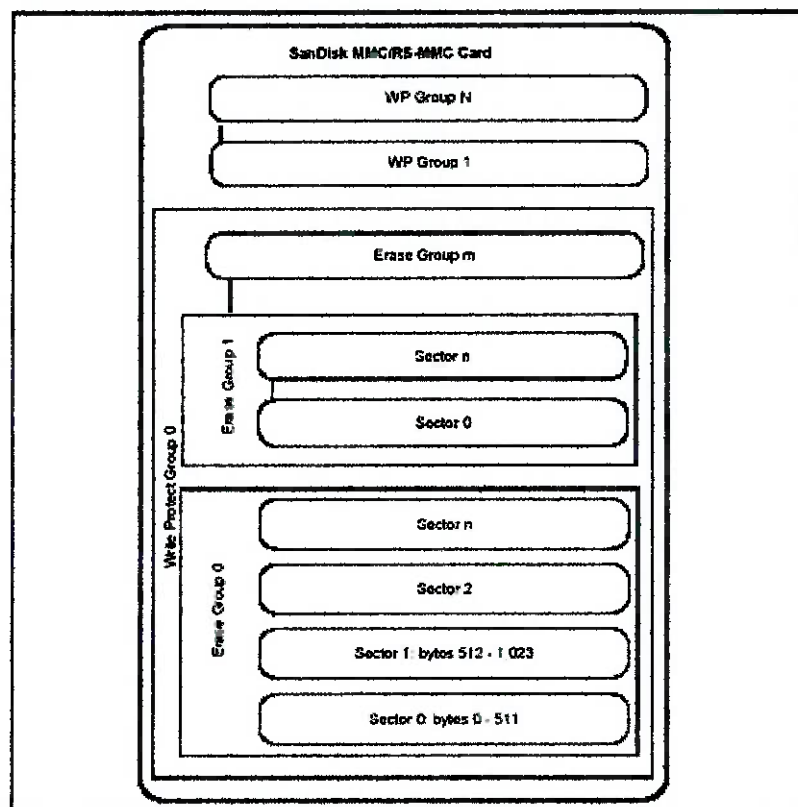


Figure 1-3 Data Transfer Formats

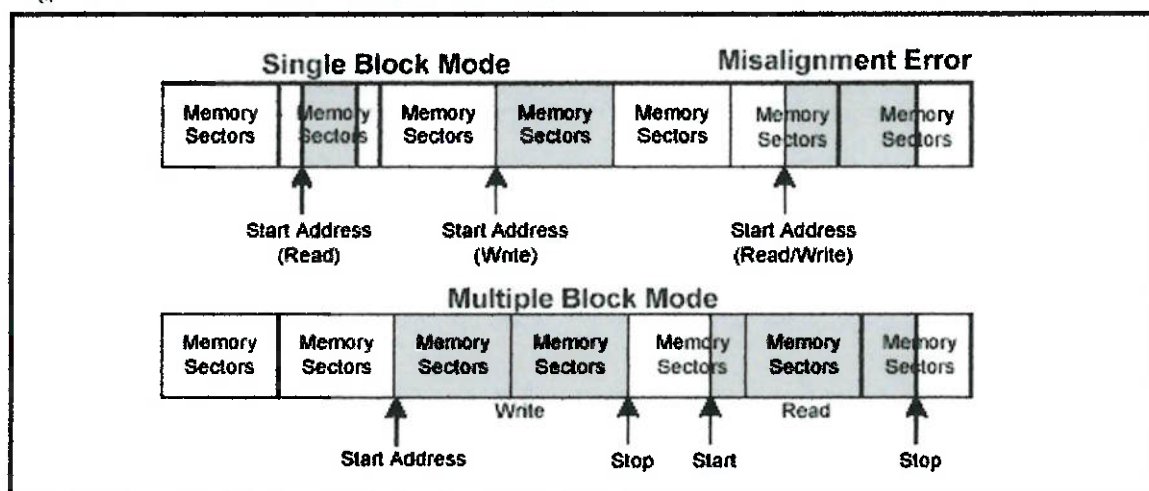


Table 1-2 Mode Definitions

Mode	Description
Single Block	In this mode the host reads or writes one data block in a pre-specified length. The data block transmission is protected with 16-bit CRC that is generated by the sending unit and checked by the receiving unit. The block length for read operations is limited by the device sector size (512 bytes) but can be as small as a single byte. Misalignment is not allowed. Every data block must be contained in a single physical sector. The block length for write operations must be identical to the sector size and the start address aligned to a sector boundary.
Multiple Block	This mode is similar to the single block mode, except for the host can read/write multiple data blocks (all have the same length) that are stored or retrieved from contiguous memory addresses starting at the address specified in the command. The operation is terminated with a stop transmission command. Misalignment and block length restrictions apply to multiple blocks and are identical to the single block read/write operations.

Table 2-1 Environmental Specification Summary

Temperature	Operating	-25° C to 85° C
	Non-operating	-40° C to 85° C
Humidity	Operating	8% to 95%, non condensing
	Non-operating	8% to 95%, non condensing
Acoustic Noise		0 dB
ESD Protection	Contact Pads	+/- 4kV. Human body model according to ANSI EOS/ESD-S5.1-1998
	Non Contact Pad Area	+/- 8kV (coupling plane discharge) +/- 15kV (air discharge) Human body model per IEC61000-4-2.
Vibration	Operating	15 G peak-to-peak max.
	Non-operating	15 G peak-to-peak max.
Shock	Operating	1,000 G max.
	Non-operating	1,000 G max.
Altitude¹	Operating	80,000 ft. max.
	Non-operating	80,000 ft. max.

Table 2-2 System Power Requirements

Operation	Max. Power Dissipation @3.6 V
Read	50 mA
Write	60 mA
Sleep	150 μ A

Table 2-3 Performance

Timing	Typical	Maximum
Block Read Access Time	0.5 ms	100 ms
Block Write Access Time	0.5 ms	240 ms
CMD1 to Ready after Power-up	50 ms	500 ms
Sleep to Ready	1 ms	2 ms

Table 2-4 Reliability and Maintenance Summary

MTBF	>1,000,000 hours
Preventative Maintenance	None
Data Reliability	<1 non-recoverable error in 10^{14} bits read
Endurance	100,000 write and erase cycles (typical)

SPI Mode

1	CS	I	Chip Select (active low)
2	DataIn	I	Host-to-card Commands and Data
3	VSS1	S	Supply Voltage Ground
4	VDD	S	Supply Voltage
5	CLK	I	Clock
6	VSS2	S	Supply Voltage Ground
7	DataOut	O	Card-to-host Data and Status

Figure 3-3 Power-up Diagram

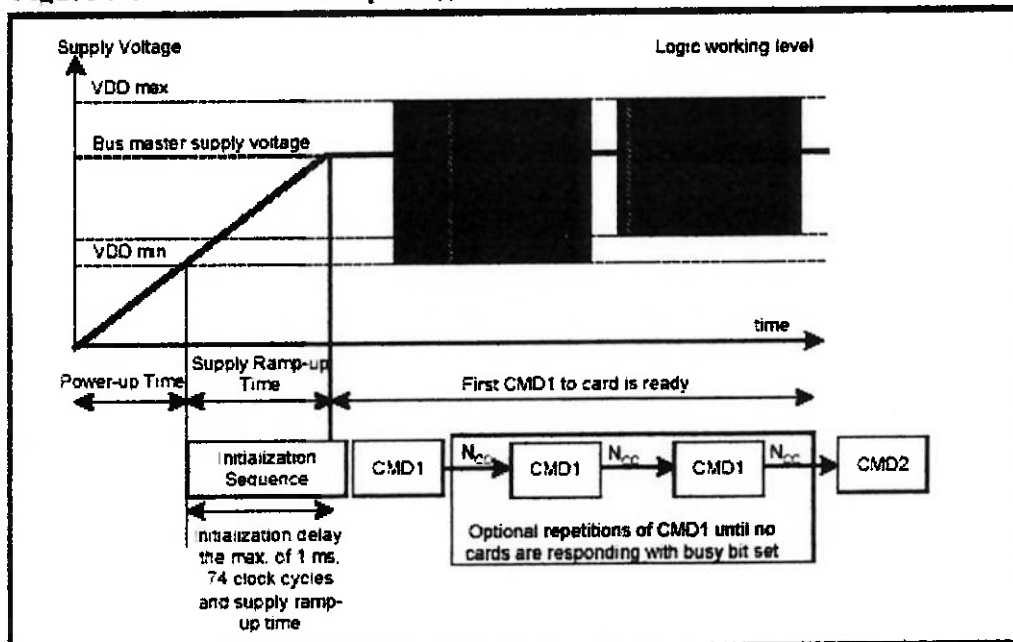


Figure 3-5 Data In/Out Referenced to Clock Timing

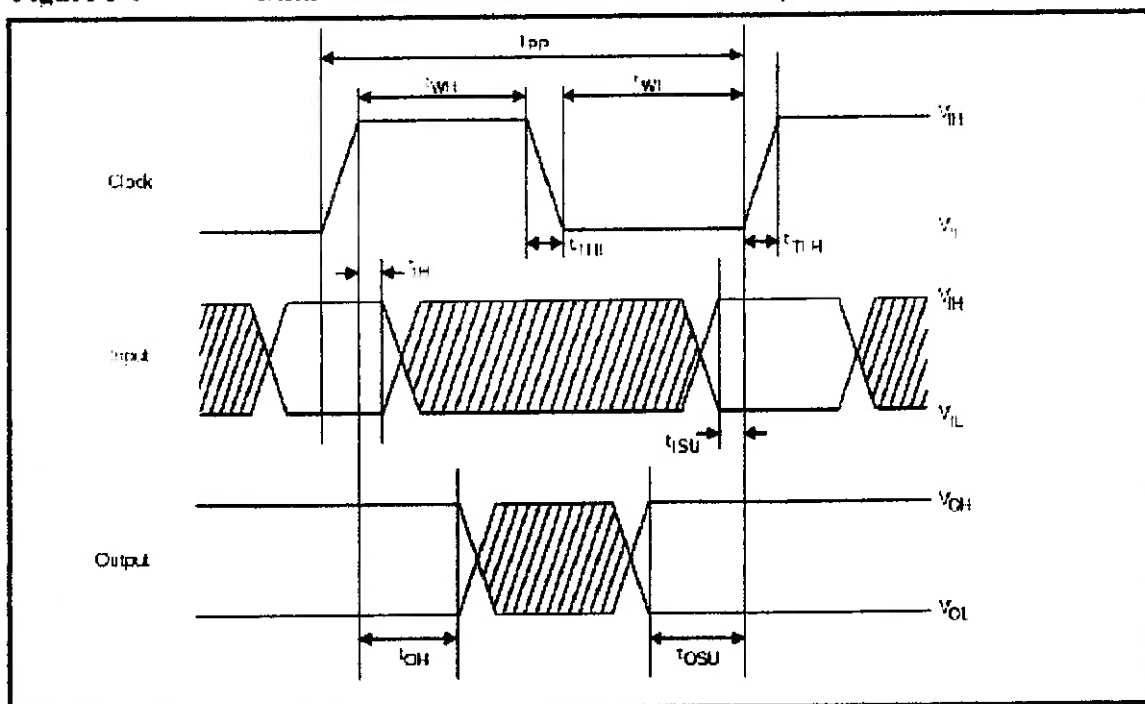


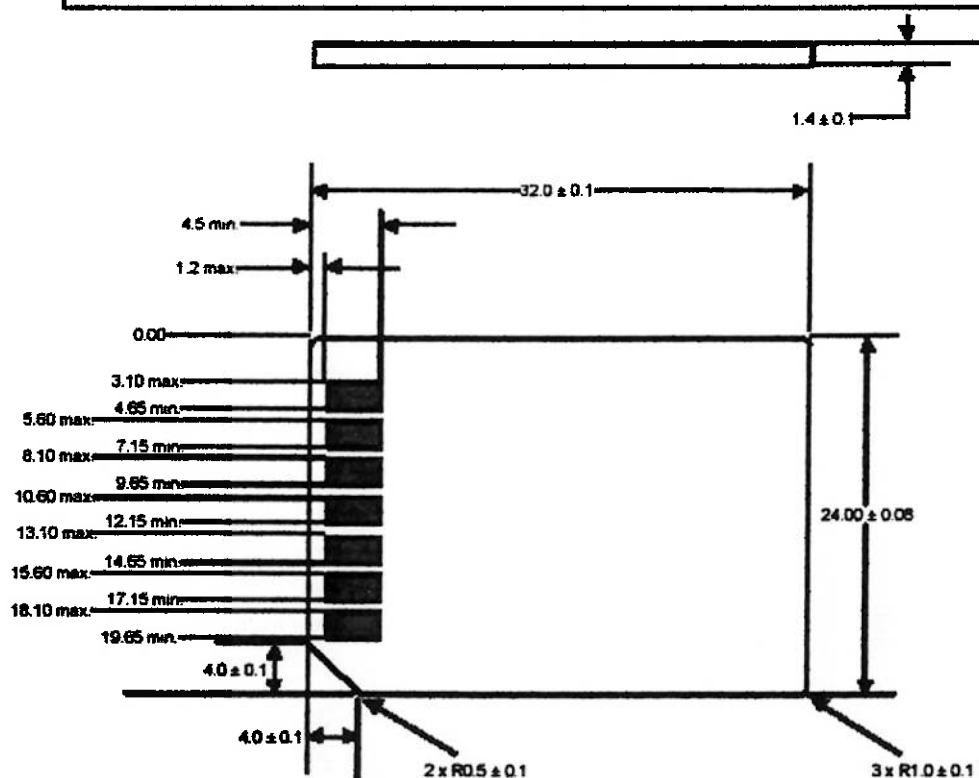
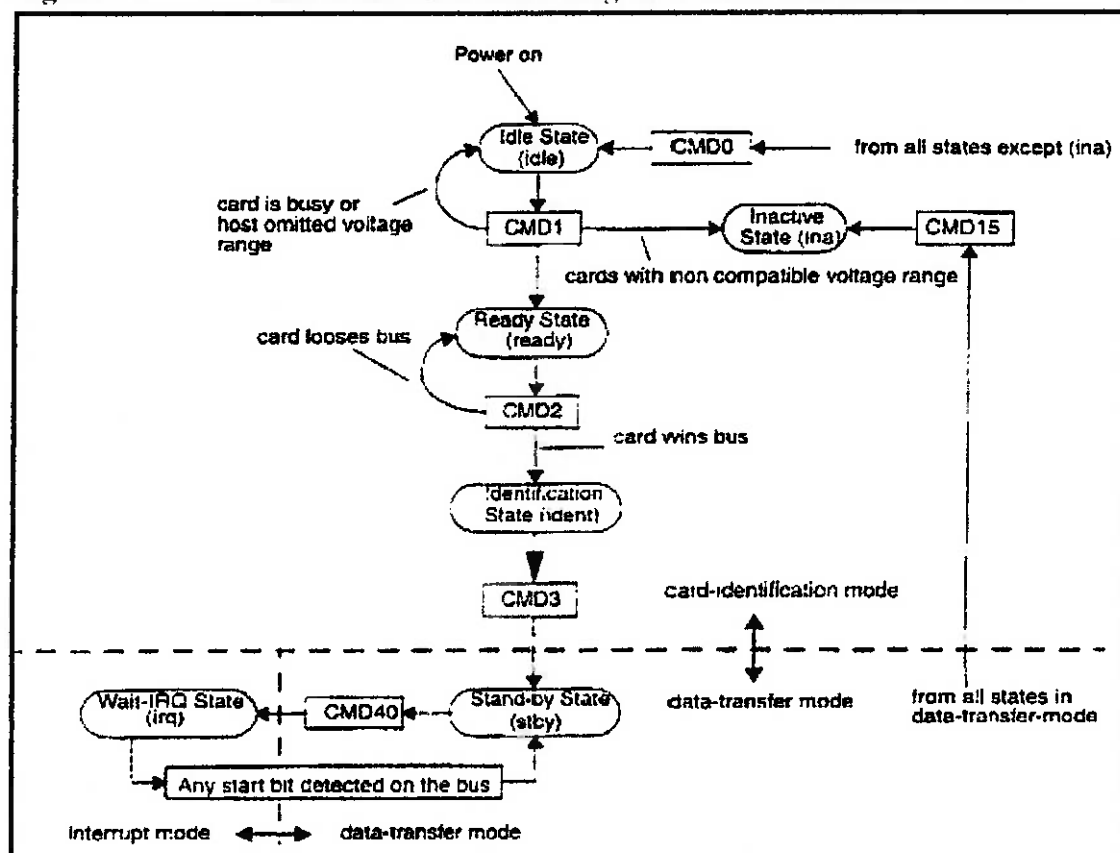
Table 3-3 Bus Operating Conditions Summary

Parameter	Symbol	Min	Max	Unit	Remark
General					
Peak voltage on all lines	---	-0.5	3.6	V	
All Inputs					
Input Leakage Current	---	-10	10	uA	
All Outputs					
Output Leakage Current	---	-10	10	uA	
Power Supply Voltage³					
Supply Voltage	V _{DD}	2.7	3.6	V	
Supply voltage differentials (V _{SS1} , V _{SS2})	---	-0.5	0.5	V	
Capacitance					
V _{DD} Capacitance	C (V _{DD})	---	3.0	uF	

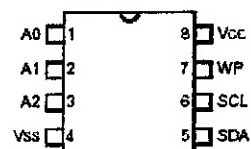
Table 3-7 Bus Timing

Parameter	Symbol	Min	Max	Unit	Remark
Clock (CLK) – all values referred to min. V_M and max. V_L					
Clock Freq. Data Transfer Mode	f _{PP}	0	20	MHz	C _L ≤ 100 pF (10 cards)
Clock Freq. Identification Mode	f _{OD}	0	400	kHz	C _L ≤ 250 pF (30 cards)
Clock Low Time	t _{WL}	10	---	ns	C _L ≤ 100 pF (10 cards)
Clock High Time	t _{WH}	10	---	ns	C _L ≤ 100 pF (10 cards)
Clock Rise Time	t _{TLH}	---	10	ns	C _L ≤ 100 pF (10 cards)
Clock Fall Time	t _{THL}	---	10	ns	C _L ≤ 100 pF (10 cards)
Clock Low Time	t _{WL}	50	---	ns	C _L ≤ 250 pF (30 cards)
Clock High Time	t _{WH}	50	---	ns	C _L ≤ 250 pF (30 cards)
Clock Rise Time	t _{TLH}	---	50	ns	C _L ≤ 250 pF (30 cards)
Clock Fall Time	t _{THL}	---	50	ns	C _L ≤ 250 pF (30 cards)
Inputs CMD, DAT – referenced to CLK					
Input setup time	t _{ISU}	3	---	ns	
Input hold time	t _{IH}	3	---	ns	
Outputs CMD, DAT – referenced to CLK					
Output setup time	t _{OSU}	5	---	ns	
Output hold time	t _{OLDV}	5	---	ns	

Figure 4-1 MultiMediaCard State Diagram—Card Identification Mode



C1. Memória 24LC04B Microchip



Memória 24LC04B

Absolute Maximum Ratings ^(†)

V _{CC}	6.5V
All inputs and outputs w.r.t. V _{SS}	-0.3V to V _{CC} +1.0V
Storage temperature.....	-65 °C to +150 °C
Ambient temperature with power applied.....	-65 °C to +125 °C
ESD protection on all pins.....	≥ 4 kV

TABLE 1-1: DC CHARACTERISTICS

DC CHARACTERISTICS			Industrial (I): TA = -40 °C to +85 °C, V _{CC} = +1.8V to +5.5V Automotive (E): TA = -40 °C to +125 °C, V _{CC} = +2.5V to +5.5V				
Param. No.	Sym.	Characteristic	Min.	Typ.	Max.	Units	Conditions
D1	V _{IH}	WP, SCL and SDA pins	—	—	—	—	—
D2	—	High-level input voltage	0.7 V _{CC}	—	—	V	—
D3	V _{IL}	Low-level input voltage	—	—	0.3 V _{CC}	V	—
D4	V _{HYS}	Hysteresis of Schmitt Trigger inputs	0.05 V _{CC}	—	—	V	(Note)
D5	V _{OL}	Low-level output voltage	—	—	0.40	V	I _{OL} = 3.0 mA, V _{CC} = 2.5V
D6	I _{LI}	Input leakage current	—	—	±1	μA	V _{IN} = V _{SS} or V _{CC}
D7	I _{LO}	Output leakage current	—	—	±1	μA	V _{OUT} = V _{SS} or V _{CC}
D8	C _{IN} , C _{OUT}	Pin capacitance (all inputs/outputs)	—	—	10	pF	V _{CC} = 5.0V (Note) TA = 25 °C, F _{CLK} = 1 MHz
D9	I _{CC} write	Operating current	—	0.1	3	mA	V _{CC} = 5.5V, SCL = 400 kHz
D10	I _{CC} read		—	0.05	1	mA	—
D11	I _{CCS}	Standby current	—	0.01	1	μA	Industrial
			—	—	5	μA	Automotive SDA = SCL = V _{CC} WP = V _{SS}

Note: This parameter is periodically sampled and not 100% tested.

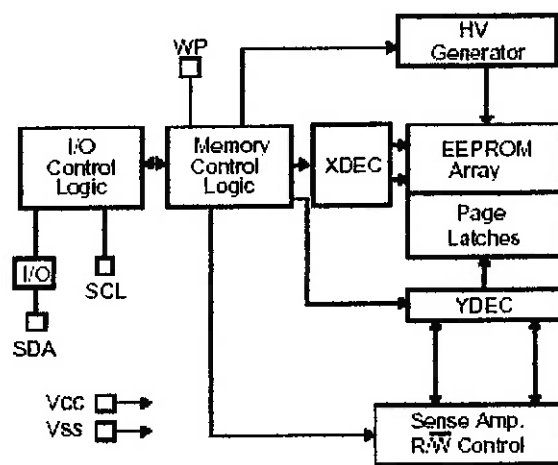
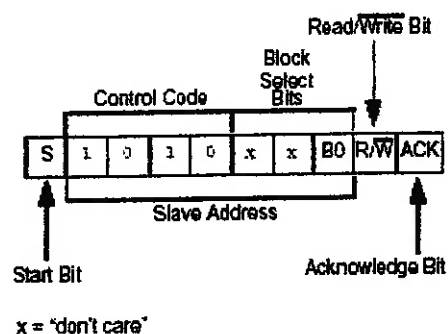


TABLE 1-2: AC CHARACTERISTICS

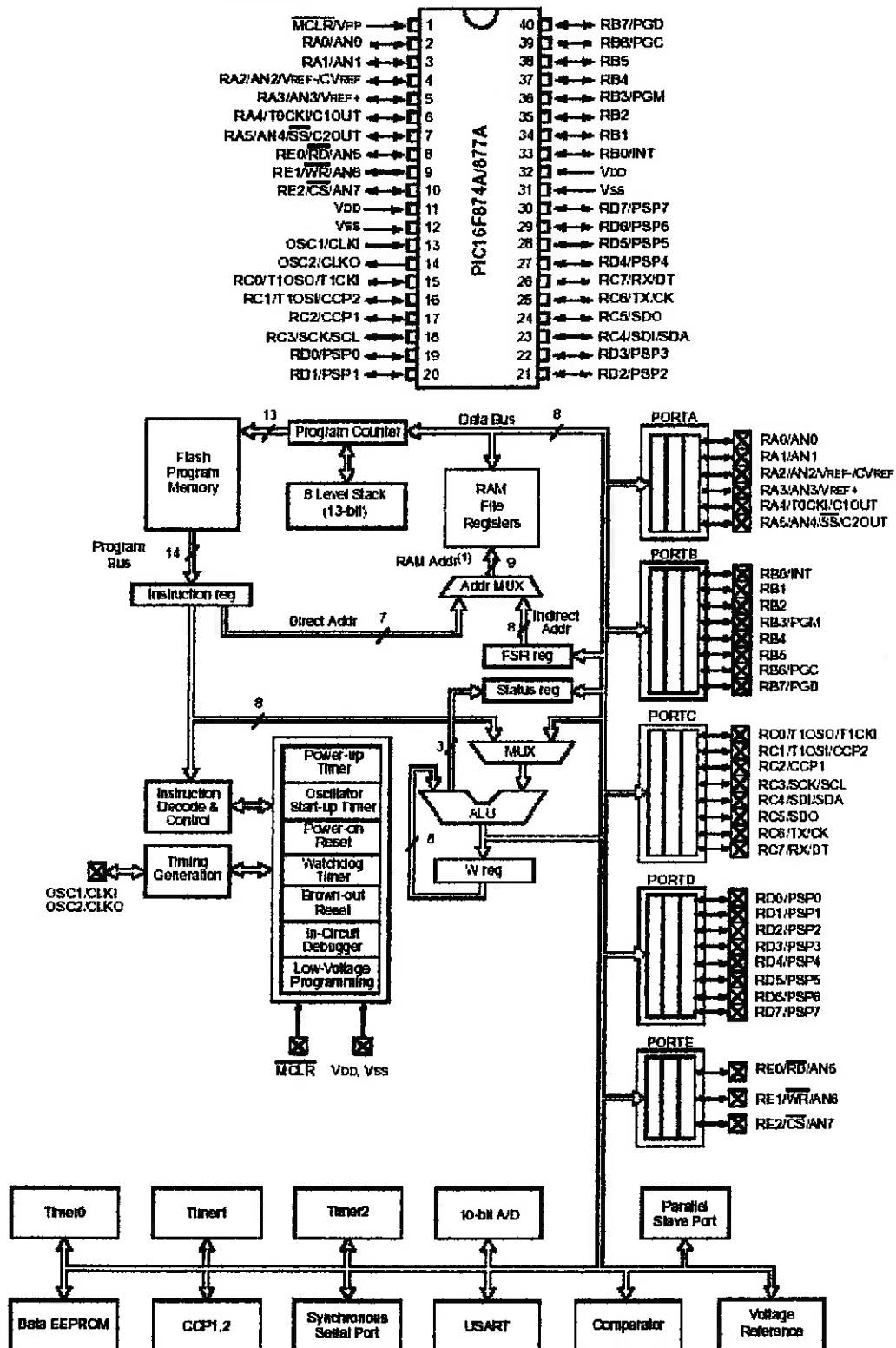
AC CHARACTERISTICS			Industrial (I): Automotive (E):		TA = -40 °C to +85 °C, VCC = +1.8V to +5.5V TA = -40 °C to +125 °C, VCC = +2.5V to +5.5V		
Param. No.	Sym.	Characteristic	Min.	Typ.	Max.	Units	Conditions
1	FCLK	Clock frequency	— —	— —	400 100	kHz	2.5V ≤ VCC ≤ 5.5V 1.8V ≤ VCC ≤ 2.5V (24AA04)
2	THIGH	Clock high time	600 4000	— —	— —	ns	2.5V ≤ VCC ≤ 5.5V 1.8V ≤ VCC ≤ 2.5V (24AA04)
3	TLOW	Clock low time	1300 4700	— —	— —	ns	2.5V ≤ VCC ≤ 5.5V 1.8V ≤ VCC ≤ 2.5V (24AA04)
4	TR	SDA and SCL rise time (Note 1)	— —	— —	300 1000	ns	2.5V ≤ VCC ≤ 5.5V (Note 1) 1.8V ≤ VCC ≤ 2.5V (24AA04) (Note 1)
5	TF	SDA and SCL fall time	— —	— —	300 —	ns	(Note 1)
6	THD:STA	Start condition hold time	600 4000	— —	— —	ns	2.5V ≤ VCC ≤ 5.5V 1.8V ≤ VCC ≤ 2.5V (24AA04)
7	TSU:STA	Start condition setup time	600 4700	— —	— —	ns	2.5V ≤ VCC ≤ 5.5V 1.8V ≤ VCC ≤ 2.5V (24AA04)
8	THD:DAT	Data input hold time	0 —	— —	— —	ns	(Note 2)
9	TSU:DAT	Data input setup time	100 250	— —	— —	ns	2.5V ≤ VCC ≤ 5.5V 1.8V ≤ VCC ≤ 2.5V (24AA04)
10	TSU:STO	Stop condition setup time	600 4000	— —	— —	ns	2.5V ≤ VCC ≤ 5.5V 1.8V ≤ VCC ≤ 2.5V (24AA04)
11	TAA	Output valid from clock (Note 2)	— —	— —	900 3500	ns	2.5V ≤ VCC ≤ 5.5V 1.8V ≤ VCC ≤ 2.5V (24AA04)
12	TBUF	Bus free time: Time the bus must be free before a new transmission can start	1300 4700	— —	— —	ns	2.5V ≤ VCC ≤ 5.5V 1.8V ≤ VCC ≤ 2.5V (24AA04)
13	TOF	Output fall time from VIH minimum to VIL maximum	20+0.1CB —	— —	250 250	ns	2.5V ≤ VCC ≤ 5.5V 1.8V ≤ VCC ≤ 2.5V (24AA04)
14	TSP	Input filter spike suppression (SDA and SCL pins)	— —	— —	50 —	ns	(Notes 1 and 3)
15	TWC	Write cycle time (byte or page)	— —	— —	5 —	ms	—
16	—	Endurance	1M —	— —	— —	cycles	25 °C (Note 4)

Operation	Control Code	Block Select	R/W
Read	1010	Block Address	1
Write	1010	Block Address	0



ANEXO D – Microcontrolador PIC16F877A Microchip

40-Pin PDIP



**FIGURE 2-1: PIC16F876A/877A
PROGRAM MEMORY MAP
AND STACK**

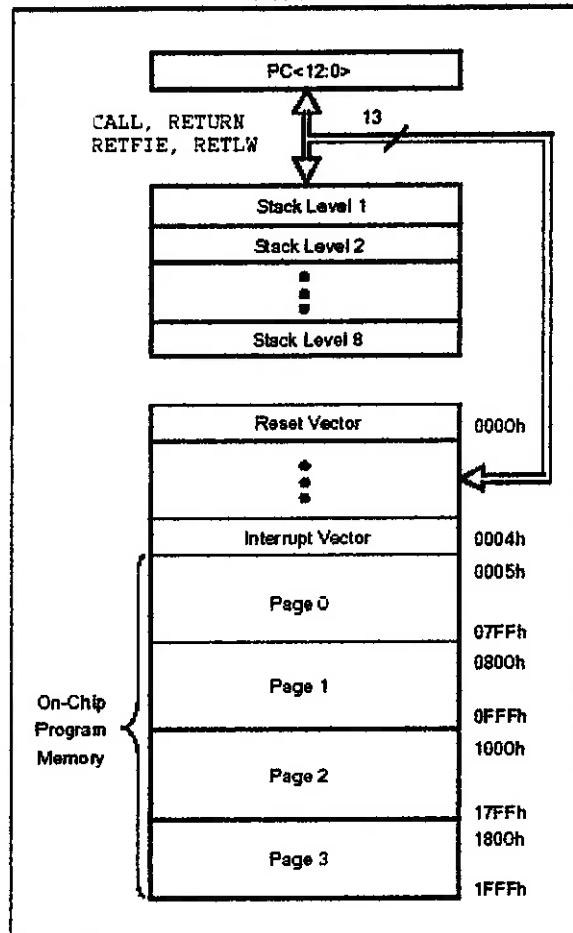


FIGURE 6-2: TIMER1 BLOCK DIAGRAM

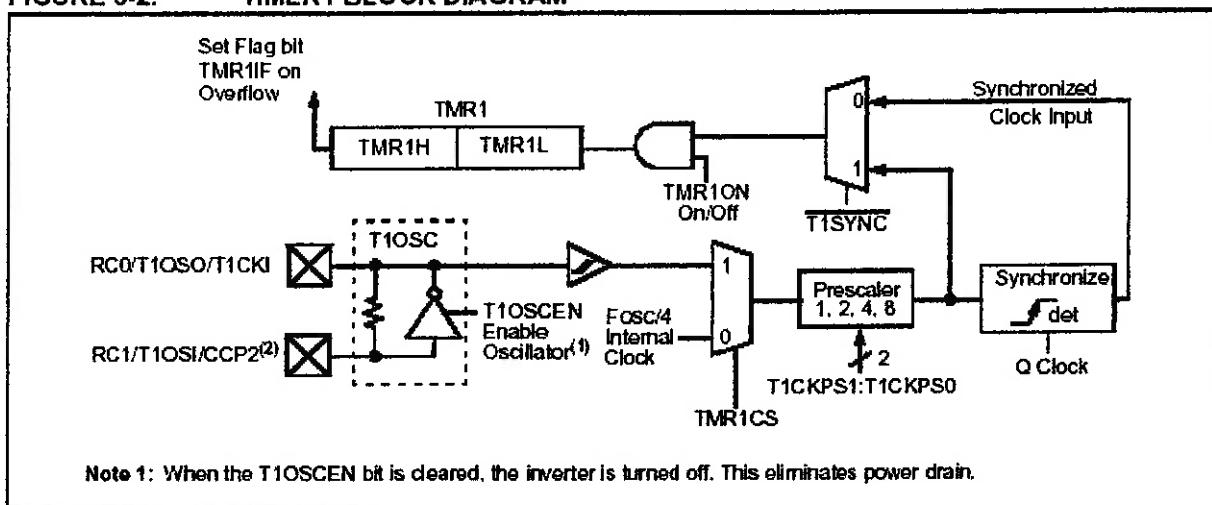


FIGURE 10-6: USART RECEIVE BLOCK DIAGRAM

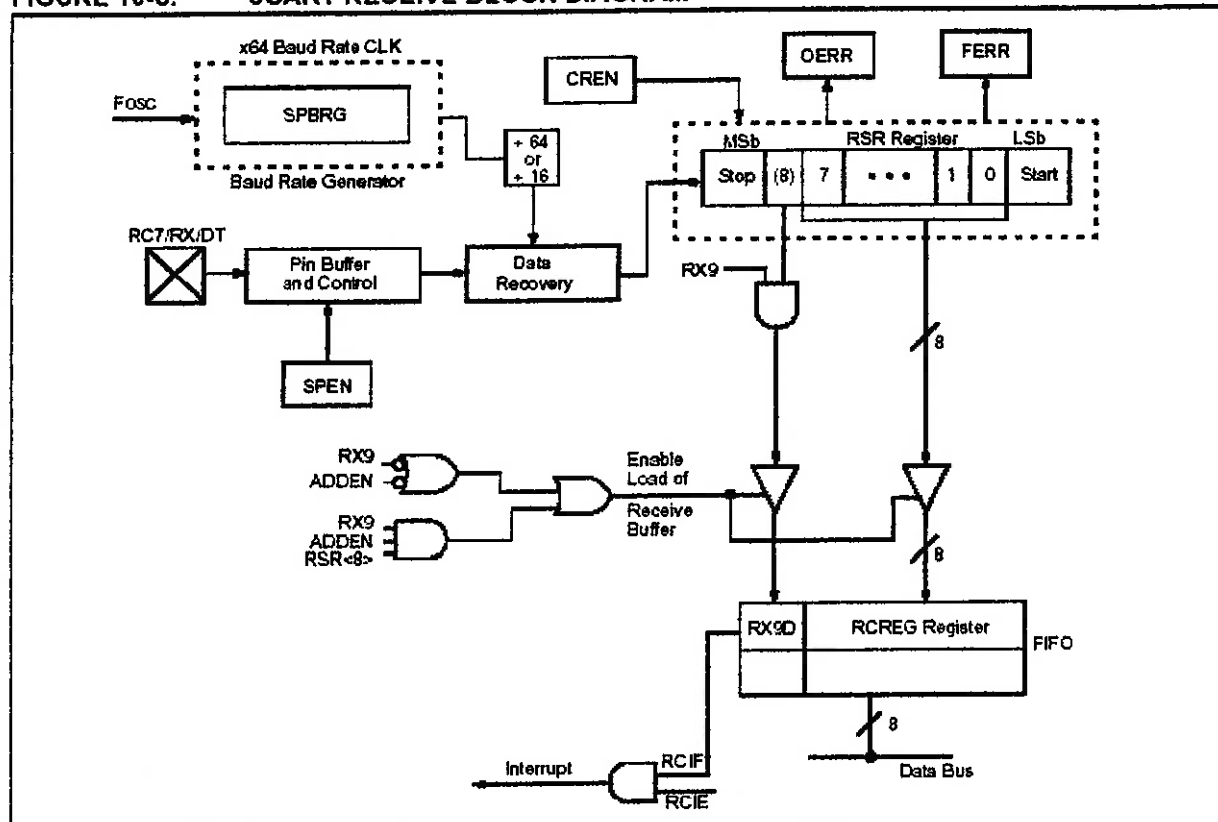


FIGURE 14-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)

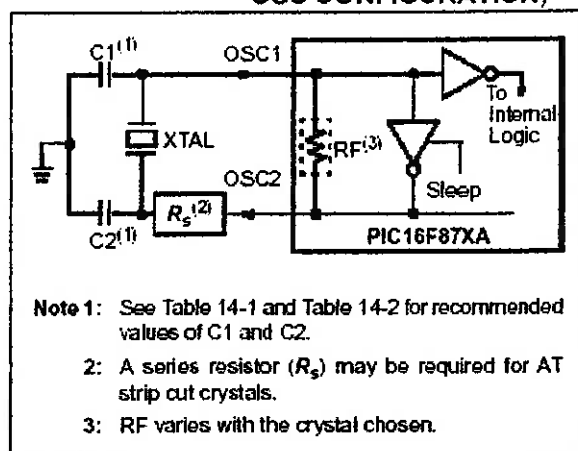


TABLE 14-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq.	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

FIGURE 18-1: TYPICAL I_{DD} vs. F_{OSC} OVER V_{DD} (HS MODE)

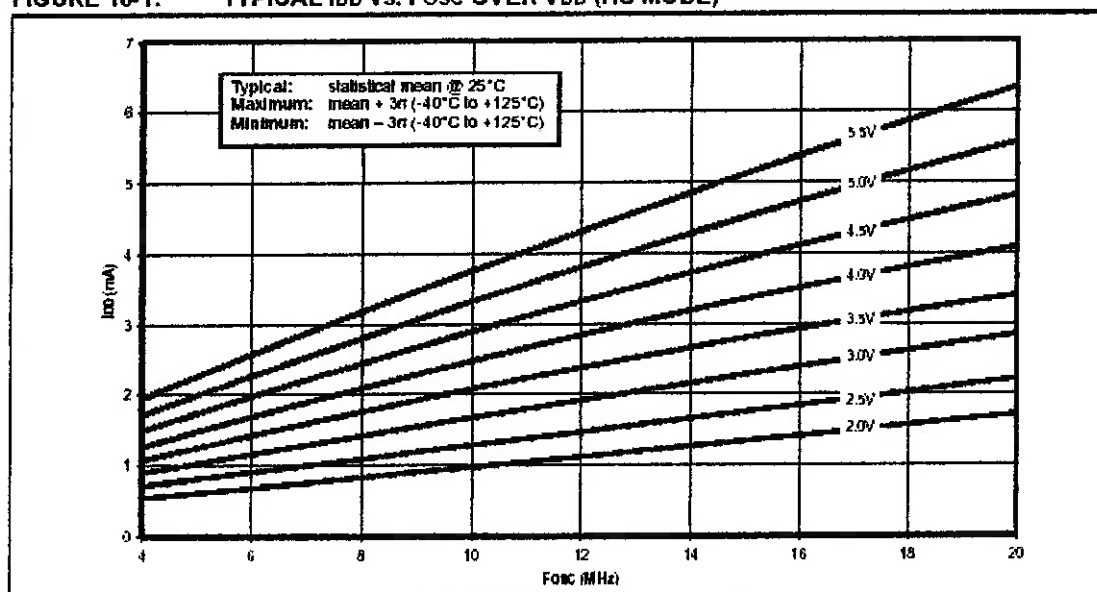
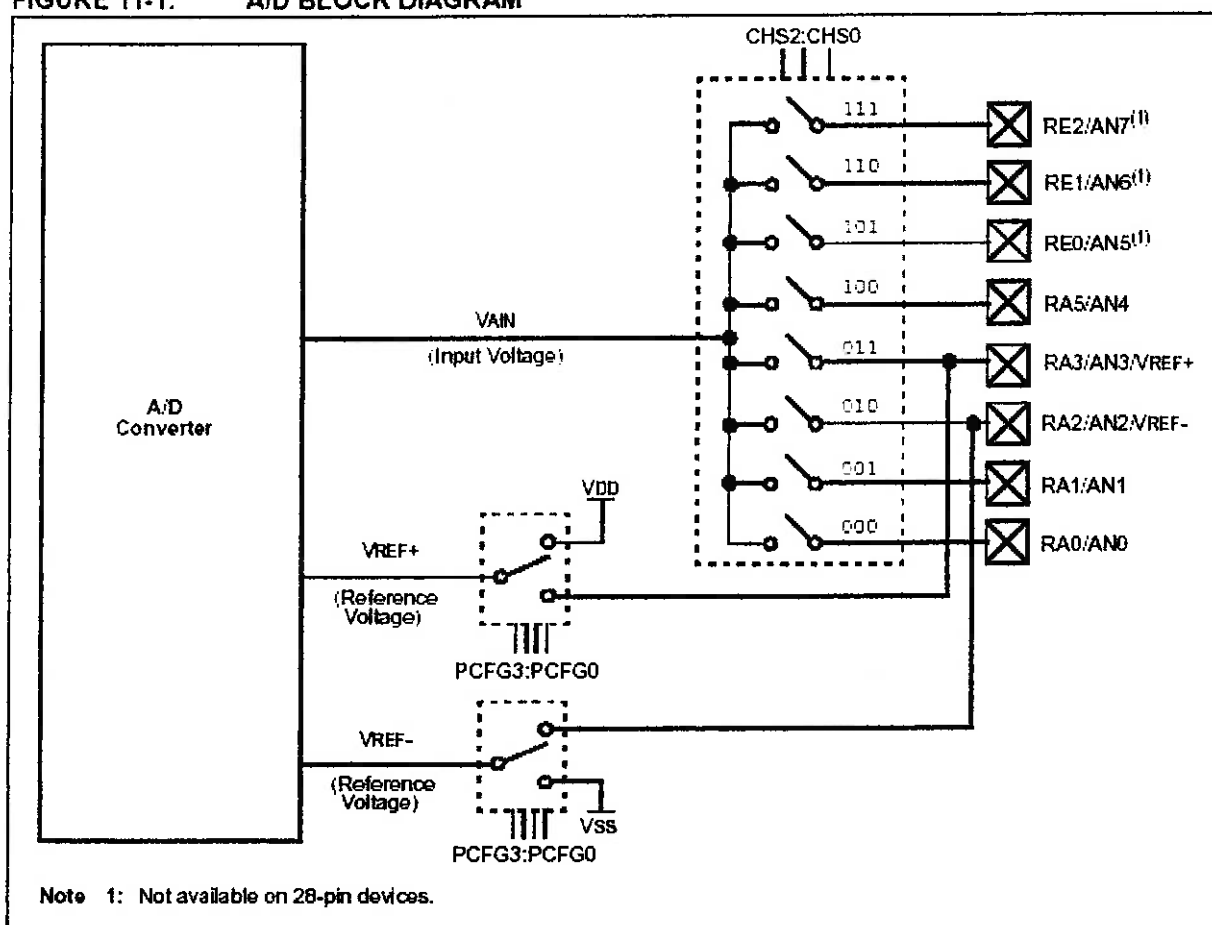


FIGURE 11-1: A/D BLOCK DIAGRAM



Absolute Maximum Ratings †

Ambient temperature under bias	-55 to +125 °C
Storage temperature	-65 °C to +150 °C
Voltage on any pin with respect to VSS (except VDD, MCLR, and RA4)	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to VSS	-0.3 to +7.5V
Voltage on MCLR with respect to VSS (Note 2)	0 to +14V
Voltage on RA4 with respect to VSS	0 to +8.5V
Total power dissipation (Note 1)	1.0W
Maximum current out of VSS pin	300 mA
Maximum current into VDD pin	250 mA
Input clamp current, I _{IK} (V _I < 0 or V _I > VDD)	± 20 mA
Output clamp current, I _{OK} (V _O < 0 or V _O > VDD)	± 20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by PORTA, PORTB and PORTE (combined) (Note 3)	200 mA
Maximum current sourced by PORTA, PORTB and PORTE (combined) (Note 3)	200 mA
Maximum current sunk by PORTC and PORTD (combined) (Note 3)	200 mA
Maximum current sourced by PORTC and PORTD (combined) (Note 3)	200 mA

TABLE 17-14: A/D CONVERTER CHARACTERISTICS: PIC16F873A/874A/876A/877A (INDUSTRIAL)
PIC16LF873A/874A/876A/877A (INDUSTRIAL)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
A01	Nr	Resolution	—	—	10-bits	bit	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A03	EIL	Integral Linearity Error	—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A04	EDL	Differential Linearity Error	—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A06	E _{OFF}	Offset Error	—	—	< ± 2	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A07	E _{GN}	Gain Error	—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A10	—	Monotonicity	—	guaranteed ³⁾	—	—	VSS ≤ VAIN ≤ VREF
A20	VREF	Reference Voltage (VREF+ – VREF-)	2.0	—	VDD + 0.3	V	
A21	VREF+	Reference Voltage High	AVDD – 2.5V	—	AVDD + 0.3V	V	
A22	VREF-	Reference Voltage Low	AVSS – 0.3V	—	VREF+ – 2.0V	V	
A25	VAIN	Analog Input Voltage	VSS – 0.3V	—	VREF + 0.3V	V	
A30	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	2.5	kΩ	(Note 4)
A40	IAD	A/D Conversion Current (VDD)	PIC16F87XA	—	220	—	Average current consumption when A/D is on (Note 1)
			PIC16LF87XA	—	90	—	
A50	IREF	VREF Input Current (Note 2)	—	—	5	μA	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 11.1 "A/D Acquisition Requirements". During A/D conversion cycle
			—	—	150	μA	

FIGURE 17-19: A/D CONVERSION TIMING

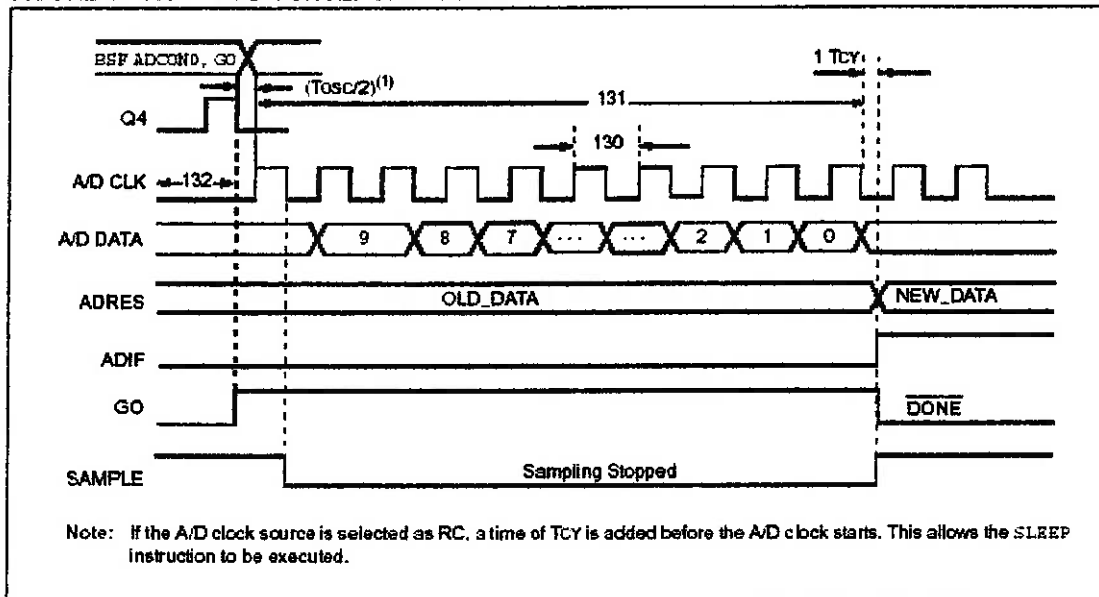


TABLE 17-15: A/D CONVERSION REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D Clock Period	PIC16F87XA	1.6	—	—	μs	TOSC based, VREF ≥ 3.0V
			PIC16LF87XA	3.0	—	—	μs	TOSC based, VREF ≥ 2.0V
			PIC16F87XA	2.0	4.0	6.0	μs	A/D RC mode
			PIC16LF87XA	3.0	6.0	9.0	μs	A/D RC mode
131	TCNV	Conversion Time (not including S/H time) (Note 1)		—	12	TAD		
132	TACQ	Acquisition Time	(Note 2)	40	—	μs	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSB (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).	
			10*	—	—	μs		
134	TGO	Q4 to A/D Clock Start	—	TOSC/2 §	—	—	If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification ensured by design.

Note 1: ADRES register may be read on the following TCY cycle.

2: See Section 11.1 "A/D Acquisition Requirements" for minimum conditions.

ANEXO E – Bibliotecas do Compilador MikroC MikroElektronika e Código Fonte

```
/*
* Nome do Projeto:
  Sistema de Aquisicao de Dados Embarcado em Aeronave
* Copyright:
  Gabriel Severino da Silva Junior, 2005.
* Descricao:
  Este programa foi realizado como parte do Projeto de
  Formatura entregue na Escola Politecnica da USP
* Configuracao:
  MCU:      P16F877A
  Placa:     SADEA v1.0b
  Oscilador: HS, 20 MHz
  Modulos:   -
  Compilador: MikroC v2.1.0.0
* Observacoes:
  Nenhuma.
*/

#define BUFFER_SIZE 8

// Variaveis globais
unsigned short chaves, cnt;
unsigned short buffer_aux[BUFFER_SIZE][BUFFER_SIZE];
unsigned long endereco;

// Le portas analogicas e preenche buffer k na coluna j
void F_CarregaBuffer(unsigned short j)
{
  unsigned short i;
  unsigned int temp_aux;
  temp_aux = 0;
  for (i = 0; i<BUFFER_SIZE-2; i++)
  {
    temp_aux = Adc_Read(i);
    buffer_aux[i][j] = (temp_aux >> 2);
  }
  buffer_aux[BUFFER_SIZE-2][j] = TMR1L;
  buffer_aux[BUFFER_SIZE-1][j] = cnt; //ultimo byte do buffer armazena a contagem
  TMR1L=0x00;
} // Fim da funcao
```

```

// Envia comando para o MMC e recebe resposta
// Retorna a resposta R1 do MMC
short F_ComandoMmc(char comando, unsigned long argumento)
{
    short aux;
    aux = 0;
    Spi_Write(0xFF); // Ncs
    Spi_Write(comando);
    Spi_Write(argumento >> 24);
    Spi_Write(argumento >> 16);
    Spi_Write(argumento >> 8);
    Spi_Write(argumento);
    Spi_Write(0xFF); // Bit de parada, com CRC inativo
    Spi_Write(0xFF); // Ncr
    aux = Spi_Read(0xFF); // Resposta ao comando
    Spi_Write(0xFF); // Nwr, Nac, Nec
    return aux;
} // Fim da funcao

```

```

// Re-inicia MMC
void F_ResetMmc()
{
    F_ComandoMmc(0x40, 0); // CMD0
    while (F_ComandoMmc(0x41, 0) != 0); // CMD1
}

```

```

// Inicia e configura MMC (configuracao default, 512kb, CRC OFF, ...)
void F_IniciaMmc()
{
    unsigned short i;
    i = 0; PORTC.F2 = 1; //CS# = 1
    for (i=0; i<10; i++) // Inicia MMC
    {
        Spi_Write(0xFF);
    }
    PORTC.F2 = 0; //CS# = 0
    F_ResetMmc();
}

```

```

// Retorna 0 se MMC esta pronto, 1 se MMC esta ocupado
short F_StatusMmc()
{
    short aux1, aux2, saida;
    saida=1;
    Spi_Write(0xFF); // Ncs
    Spi_Write(0x4D); //CMD13
    Spi_Write(0x00);
}

```

```

Spi_Write(0x00);
Spi_Write(0x00);
Spi_Write(0x00);
Spi_Write(0xFF); // Bit de parada. com CRC inativo
Spi_Write(0xFF); // Ncr
aux1 = Spi_Read(0xFF); // Resposta ao comando
aux2 = Spi_Read(0xFF); // Resposta ao comando
if ( (aux1==0x00)&(aux2==0x00) ) saida=0;
return saida;
} // Fim da funcao

```

```

// Envia buffer para o MMC. Se finaliza igual a 1,
// retorna a resposta, caso contrario retorna 0
short F_EnviaMmc(short finaliza)
{
short i, j, aux;
aux = 0;
if (finaliza == 0) Spi_Write(0xFE); // Byte de inicio
for (j = 0; j<BUFFER_SIZE; j++)
{
for (i = 0; i<BUFFER_SIZE; i++)
{
Spi_Write(buffer_aux[i][j]);
}
}
if (finaliza == 1)
{
Spi_Write(0xFF); // CRC dos dados
Spi_Write(0xFF); // CRC dos dados, fim do token
aux = Spi_Read(0xFF); // Resposta aos dados
Spi_Write(0xFF); //Nec
}
return aux;
} // Fim da funcao

```

```

// Recebe buffer do MMC
void F_RecebeMmc(short finaliza)
{
short i, j;
if (finaliza == 0) Spi_Read(0xFF); // Byte de inicio
for (j = 0; j<BUFFER_SIZE; j++)
{
for (i = 0; i<BUFFER_SIZE; i++)
{
buffer_aux[i][j] = Spi_Read(0xFF);
}
}
}

```

```

if (finaliza == 1)
{
    Spi_Read(0xFF); // CRC dos dados
    Spi_Read(0xFF); // CRC dos dados, fim do token
    Spi_Write(0xFF); //Nec
}
} // Fim da funcao

// Verifica a configuracao das chaves
void F_LeChaves()
{
    chaves = 0;
    if (PORTD.F3 == 1) chaves +=1;
    if (PORTD.F2 == 1) chaves +=2;
    PORTC.F0 = ~(PORTD.F3 | PORTD.F2);
    PORTD.F0 = PORTD.F3;
    PORTD.F1 = PORTD.F2;
} // Fim da funcao

// Escreve o buffer_aux na memoria 24LC04
void F_EscreveMem(short addr, short bloco)
{
    unsigned short i, j;
    INTCON = 0x00; // Desabilita interrupcoes
    for (j = 0; j<BUFFER_SIZE; j++)
    {
        Soft_I2C_start(); // Envia sinal de inicio
        if (bloco==0) Soft_I2C_Write(0xA0); // Envia comando
        if (bloco==1) Soft_I2C_Write(0xA2); // Envia comando
        Soft_I2C_Write(addr); // Envia endereco
        for (i = 0; i<BUFFER_SIZE; i++)
        {
            Soft_I2C_Write(buffer_aux[i][j]); // Envia dados
        }
        Soft_I2C_Stop(); // Envia sinal de parada
        addr += 8;
    }
    INTCON = 0xA0; // Habilita interrupcao do TMR0
} // Fim da funcao

// Le a memoria 24LC04 e carrega o buffer_aux
void F_LeMem(short addr, short bloco)
{
    unsigned short i, j;
    INTCON = 0x00; // Desabilita interrupcoes
    for (j = 0; j<BUFFER_SIZE; j++)
    {

```

```

Soft_I2C_start(); // Envia sinal de inicio
if (bloco==0) Soft_I2C_Write(0xA1); // Envia comando
if (bloco==1) Soft_I2C_Write(0xA3); // Envia comando
Soft_I2C_Write(addr); // Envia endereco
Soft_I2C_Start(); // Envia sinal de inicio
Soft_I2C_Write(0xA3); // Envia comando
for (i = 0; i<BUFFER_SIZE-1; i++)
{
    buffer_aux[i][j] = Soft_I2C_Read(1u); // Recebe byte com acknowledge
}
buffer_aux[BUFFER_SIZE-1][j] = Soft_I2C_Read(0u); // Recebe byte sem acknowledge
Soft_I2C_Stop(); // Envia sinal de parada
addr += 8;
}
INTCON = 0xA0; // Habilita interrupcao do TMR0
} // Fim da funcao

```

```

// Envia buffer para USART
void F_EnviaUSART()
{
    short i, j;
    for (j=0; j<BUFFER_SIZE; j++)
    {
        for (i=0; i<BUFFER_SIZE; i++)
        {
            Uart_Write(buffer_aux[i][j]);
        }
    }
} // Fim da funcao

```

```

// Modo stand by
void F_StandBy()
{
    while (chaves==0)
    {
        F_LeChaves();
        // Se MMC estiver vazio, nao pode le-lo. Permanece em stand by.
        if ((endereco==0)&&(chaves==2)) chaves=0;
        // Se houve overflow no MMC, nao pode grava-lo. Permanece em stand by.
        if ( ( endereco>=0xFFFFFBE) && ((chaves==1)|| (chaves==3)) )chaves=0;
    }
} // Fim da funcao

```

```

// Modo de leitura e comunicacao USART
void F_LeMmcComunica()
{
    short i, k;

```

```
unsigned long addr;
addr = 0;
```

```
while (F_StatusMmc());
while ( (addr<endereco) && (chaves==2) )
{
    F_ComandoMmc(0x51, addr);
    for (k=0;k<8;k++)
    {
        if (k!=7) F_RecebeMmc(0);
        if (k==7) F_RecebeMmc(1);
        F_EnviaUSART();
    }
    addr+=512;
    F_LeChaves();
}
} // Fim da funcao
```

// Ler MMC, amostrar, comunicar via USART

```
void F_AmostraArmazenaComunica()
```

```
{
    short i, amostra, addr, k;
    amostra=0;

    while ( (endereco<0xFFFFFBE) && ((chaves==1) || (chaves==3)) )
    {
        while (F_StatusMmc());
        k=0;
        F_ComandoMmc(0x58, endereco);
        addr = 0;
        for (i=0;i<8;i++) // amostra 8 buffers
        {
            amostra=0;
            if (i==4)
            {
                k=1;
                addr=0;
            }
            while (amostra<BUFFER_SIZE) // Amostra 8 vezes os sensores e carrega o buffer
            {
                if (cnt >= 25)
                {
                    F_CarregaBuffer(amostra);
                    amostra++;
                    cnt=0;
                }
            }
        }
    }
}
```

```

    if (chaves==3) F_EnviaUSART();
    F_EscreveMem(addr,k);
    addr+=64;
}
for (k=0;k<2;k++)
{
    F_LeMem(0,k);
    F_EnviaMmc(0);
    F_LeMem(64,k);
    F_EnviaMmc(0);
    F_LeMem(128,k);
    F_EnviaMmc(0);
    F_LeMem(192,k);
    if (k==0) F_EnviaMmc(0);
    if (k==1) F_EnviaMmc(1);
}
endereco+=512;
F_LeChaves();
}
} // Fim da funcao

// Interrupcao - frequencia de 50Hz para amostragem
void interrupt() {
    cnt++; // Incrementa contador
    TMR0 = 130;
    INTCON = 0x20; // Configura T0IE, e limpa T0IF
} // Fim da funcao

// Programa principal
void main()
{
    // Limpa Registradores das Portas
    // Pinos analogicos, com RE1 e RE2 Digitais
    // Configura pinos de entrada e saida
    // Configura e Inicia comunicacao SPI, USART e I2C
    // Configura TMR0, TMR1 e interrupcao
    // Configura e inicia comunicacao com MMC
    // Inicia variaveis globais
    PORTA = 0x00; PORTB = 0x00; PORTC = 0x04; PORTD = 0x00; PORTE = 0x00;
    ADCON1 = 0x89;
    TRISA = 0xFF; TRISB = 0x7F; TRISC = 0xFA; TRISD = 0xFC; TRISE = 5;
    Usart_Init(57600);
    Spi_Init_Advanced(MASTER_OSC_DIV64, DATA_SAMPLE_MIDDLE, CLK_IDLE_LOW,
    LOW_2_HIGH);
    Soft_I2C_Config(&PORTE, 2, 1);
    INTCON = 0xA0; OPTION_REG = 0x84;
    TMR0 = 130; T1CON = 0x07; TMR1L = 0x00;

```

```
F_IniciaMmc(); PORTC.F2 = 0; //CS# =0  
chaves=0; cnt=0; endereco=0;
```

```
// Piscando os LEDs
```

```
PORTC.F0 = 0; PORTD.F0 = 0; PORTD.F1 = 0; Delay_100ms();  
PORTC.F0 = 1; PORTD.F0 = 1; PORTD.F1 = 1; Delay_100ms();  
PORTC.F0 = 0; PORTD.F0 = 0; PORTD.F1 = 0; Delay_100ms();  
PORTC.F0 = 1; PORTD.F0 = 1; PORTD.F1 = 1; Delay_100ms();  
PORTC.F0 = 0; PORTD.F0 = 0; PORTD.F1 = 0; Delay_100ms();  
PORTC.F0 = 1; PORTD.F0 = 1; PORTD.F1 = 1; Delay_100ms();  
PORTC.F0 = 0; PORTD.F0 = 0; PORTD.F1 = 0; Delay_100ms();  
PORTC.F0 = 1; PORTD.F0 = 1; PORTD.F1 = 1; Delay_100ms();
```

```
while(1) // Loop eterno
```

```
{  
  F_LeChaves(); // Verifica chaves  
  switch (chaves) // Status  
  {  
    case 0: F_StandBy(); // Stand By  
    case 1: F_AmostraArmazenaComunica(); // Amost., armazen., e comun.  
    case 2: F_LeMmcComunica(); // Amostrando, e Comunindo via USART  
    case 3: F_AmostraArmazenaComunica(); // Amost., armazen., e comun.  
    default: PORTC.F0 = 1; PORTD.F0 = 1; PORTD.F1 = 1; // Erro  
  }  
}  
} // Fim do Programa
```

ADC Library

Prototype	unsigned Adc_Read(unsigned short channel);
Returns	10-bit unsigned value read from the specified channel.
Description	Initializes PIC's internal ADC module to work with RC clock. Clock determines the time period necessary for performing AD conversion (min 12TAD). Parameter channel represents the channel from which the analog value is to be acquired. Refer to the appropriate datasheet for channel-to-pin mapping.

SPI Library

Prototype	void Spi_Init(void);
Returns	Nothing.
Description	Configures and initializes SPI with default settings. Spi_Init_Advanced or Spi_Init needs to be called before using other functions from SPI Library. Default settings are: Master mode, clock Fosc/4, clock idle state low, data transmitted on low to high edge, and input data sampled at the middle of interval. For custom configuration, use Spi_Init_Advanced.
Requires	You need PIC MCU with hardware integrated SPI.

Prototype	unsigned short Spi_Read(unsigned short buffer);
Returns	Returns the received data.
Description	Provides clock by sending buffer and receives data at the end of period.
Requires	You need PIC MCU with hardware integrated SPI. SPI must be initialized and communication established before using this function. See Spi_Init_Advanced or Spi_Init.

Prototype	void Spi_Write(unsigned short data);
Returns	Nothing.
Description	Writes byte data to SSPBUF, and immediately starts the transmission.
Requires	You need PIC MCU with hardware integrated SPI. SPI must be initialized and communication established before using this function. See Spi_Init_Advanced or Spi_Init.

Prototype	void Spi_Init_Advanced(unsigned short master, unsigned short data_sample, unsigned short clock_idle, unsigned short transmit_edge);
Returns	Nothing.
Description	Configures and initializes SPI. Spi_Init_Advanced or Spi_Init needs to be called before using other functions of SPI Library. Parameter mast_slav determines the work mode for SPI; can have the following values: MASTER_OSC_DIV4 <i>// Master clock=Fosc/4</i> MASTER_OSC_DIV16 <i>// Master clock=Fosc/16</i> MASTER_OSC_DIV64 <i>// Master clock=Fosc/64</i> MASTER_TMR2 <i>// Master clock source TMR2</i> SLAVE_SS_ENABLE <i>// Master Slave select enabled</i> SLAVE_SS_DIS <i>// Master Slave select disabled</i> Parameter data_sample determines when data is sampled; can have the following values: DATA_SAMPLE_MIDDLE <i>// Input data sampled in middle of interval</i> DATA_SAMPLE_END <i>// Input data sampled at the end of interval</i> Parameter clock_idle determines idle state for clock; can have the following values: CLK_IDLE_HIGH <i>// Clock idle HIGH</i> CLK_IDLE_LOW <i>// Clock idle LOW</i> Parameter transmit_edge can have the following values: LOW_2_HIGH <i>// Data transmit on low to high edge</i> HIGH_2_LOW <i>// Data transmit on high to low edge</i>
Requires	You need PIC MCU with hardware integrated SPI.

USART Library

Prototype	void Usart_Init(const unsigned long baud_rate);
Returns	Nothing.
Description	Initializes hardware USART module with the desired baud rate. Refer to the device data sheet for baud rates allowed for specific Fosc. If you specify the unsupported baud rate, compiler will report an error.
Requires	You need PIC MCU with hardware USART. Usart_Init needs to be called before using other functions from USART Library.

Prototype	unsigned short Usart_Data_Ready(void);
Returns	Function returns 1 if data is ready or 0 if there is no data.
Description	Use the function to test if data in receive buffer is ready for reading.
Requires	USART HW module must be initialized and communication established before using this function. See Usart_Init.

Prototype	unsigned short Usart_Read(void);
Returns	Returns the received byte. If byte is not received, returns 0.
Description	Function receives a byte via USART. Use the function Usart_Data_Ready to test if data is ready first.
Requires	USART HW module must be initialized and communication established before using this function. See Usart_Init.

Prototype	void Usart_Write(unsigned short data);
Returns	Nothing.
Description	Function transmits a byte (data) via USART.
Requires	USART HW module must be initialized and communication established before using this function. See Usart_Init.

Software I²C Library

Prototype	void Soft_I2C_Config(unsigned short *port, const unsigned short SDA, const unsigned short SCL);
Returns	Nothing.
Description	Configures software I ² C. Parameter port specifies port of MCU on which SDA and SCL pins are located. Parameters SDA and SCL need to be in range 0–7 and cannot point at the same pin. Soft_I2C_Config needs to be called before using other functions from Soft I2C library.
Requires	Nothing.

Prototype	void Soft_I2C_Start(void);
Returns	Nothing.
Description	Issues START signal. Needs to be called prior to sending and receiving data.
Requires	Soft I ² C must be configured before using this function. See Soft_I2C_Config.

Prototype	unsigned short Soft_I2C_Read(unsigned short ack);
Returns	Returns one byte from the slave.
Description	Reads one byte from the slave, and sends <i>not acknowledge</i> signal if parameter ack is 0, otherwise it sends <i>acknowledge</i> .
Requires	Soft I ² C must be configured before using this function. See Soft_I2C_Config. Also, START signal needs to be issued in order to use this function. See Soft_I2C_Start.

Prototype	unsigned short Soft_I2C_Write(unsigned short data);
Returns	Returns 0 if there were no errors.
Description	Sends data byte (parameter data) via I ² C bus.
Requires	Soft I ² C must be configured before using this function. See Soft_I2C_Config. Also, START signal needs to be issued in order to use this function. See Soft_I2C_Start.

Prototype	void Soft_I2C_Stop(void);
Returns	Nothing.
Description	Issues STOP signal.
Requires	Soft I ² C must be configured before using this function. See Soft_I2C_Config.